IMPLEMENTASI *WIRELESS SENSOR NETWORK* MENGGUNAKAN *BABEL ROUTING* PROTOKOL

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Muhammad Rosyid Khulafa
NIM:125150307111012

PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

IMPLEMENTASI *WIRELESS SENSOR NETWORK* MENGGUNAKAN *BABEL ROUTING* PROTOKOL

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Muhammad Rosyid Khulafa
NIM:125150307111012

Skripsi ini telah diuji dan dinyatakan lulus pada
10 Oktober 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Sabriansyah Rizqika Akbar, S.T., M.Eng.	Gembong Edhi Setyawan, S.T., M.T.
NIP:192820809 201212 1 004	NIK:201208761201 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D.
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 19 Juli 2018

Muhammad Rosyid Khulafa

NIM: 125150307111012

KATA PENGANTAR

Puji Syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena hanya dengan rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi dengan judul "Implementasi *Wireless Sensor Network* Menggunakan *Babel Routing* Protokol". Melalui kesempatan ini, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan dan dukungan selama penulisan skripsi ini, diantaranya:

1. Bapak **Sabriansyah Rizqika Akbar, S.T., M.Eng**. selaku Dosen Pembimbing pertama yang telah membimbing dan memberikan ilmu serta saran dan perhatian dalam menyelesaikan skripsi ini.
2. Bapak **Gembong Edhi Setyawan, S.T., M.T**. selaku Dosen Pembimbing kedua yang telah membimbing dan telah memberikan ilmu serta saran dan perhatian dalam menyelesaikan skripsi ini
3. Kedua Orang Tua Penulis serta keluarga besar atas segala doa, nasihat, dukungan baik moral maupun materiil dalam melancarkan skripsi ini.
4. Abah Kyai Marzuqi Mustamar, berserta Ibu Nyai Saidah Mustamar, Mbah De Bis, Pak Lik Doel, Pak Lik Bub, Bulek Nur, Paman Zaini yang selama ini memberikan semangat baik moral maupun materiil kepada penulis dalam melancarkan skripsi.
5. Staff dosen Fakultas Ilmu Komputer, khususnya Mas Pras. Universitas Brawijaya yang telah membekali penulis berbagai ilmu selama mengikuti perkuliahan sampai akhir penulisan skripsi.
6. Kepada **Ibad** yang selama ini selalu membuatku konyol dan percaya diri, terimakasih banyak telah memberikan pukulan, tendangan, dan bantingan kepada penulis untuk melancarkan skripsi.
7. Sahabatku **Poby Zaarifwandono, M Wingga Wing, Afif Gendut, Yafi Fitza Hit dan Hasan Albana Pentol Goreng, sahabat sukhontlo group**, serta kawan kawan mahasiswa FILKOM UB angkatan 2012 atas bantuan, dukungan, motivasi dan berbagi yang turut membantu menyelesaikan skripsi ini.
8. Semua pihak yan tidak dapat penulis sebutkan satu per satu yang terlibat secara langsung atau tidak langsung agar terselesaikan skripsi ini.

Dengan segala kerendahan hati, penulis menyadari bahwa skripsi ini masih memiliki banyak kekurangan. Oleh karena itu kritik dan saran yang bersifat membangun sangat dibutuhkan sebagai pedoman untuk menyempurnakan skripsi ini agar lebih baik. Penulis berharap semoga skripsi ini dapat bermanfaat bagi diri sendiri maupun semua pihak

Malang, 19 Juli 2018

Penulis

Khulafa.profit@gmail.com

ABSTRAK

Muhammad Rosyid Khulafa, Implementasi *Wireless Sensor Network* Menggunakan *Babel Routing Protokol*

Pembimbing: Sabriansyah Rizqika Akbar, S.T., M.Eng. dan Gembong Edhi Setyawan, S.T., M.T.

Negara Indonesia memiliki wilayah yang luas memerlukan pertahanan dan strategi keamanan perang sangat diperlukan untuk mengantisipasi serangan dari berbagai wilayah. Ketika terjadi kontak perang pada wilayah wilayah tertentu, Untuk mengantisipasi peperangan, sebagai negara yang memiliki sejarah panjang haruslah memiliki miiter yang kuat. Tidak dapat dipungkiri bahwa terjadinya perang selalu menyebabkan kerusakan medan dan pos komando,disamping itu kondisi suhu dan cuaca pada medan pertempuran sangat berpengaruh terhadap kelancaran operasi. Selain itu, strategi pergerakan formasi dalam peperangan tidak hanya terfokus pada satu tempat saja,hal ini menyebabkanpenggunaan infrastruktur telekomunikasi tidak dapat maksimal. Padahal, komunikasipada saat terjadi perang sangat diperlukan agar penyampaian informasi dapat berjalan dengan baik. Perkembangan teknologi dibidang komunikasi nirkabel adalah Wireless. Yang dikembangkan saat ini menjadi *Wireless Sensor Network(WSN)*. Sebuah teknologi yang terdiri dari node-node sebagaimana tersebar dalam ruang lingkup sistem menggunakan jaringan nirkabel. Salah satu pemanfaatan teknologi *WSN* adalah untuk penerapan aplikasi agar mengetahui suhu, cuaca, jarak pada lingkungan sekitarnya, kemudian dapat bertukar data melalui jaringan node node yang telah terhubung.Seiring berjalannya waktu, meningkatnya kebutuhan yang digunakan dalam penerapan *WSN* menimbulkan permasalahan. Salah satunya jika terdapat kerusakan pada komunikasi tetap, *WSN* dapat diterapkan sebagai komunikasi sekunder atau planning lain, sebagai infrastruktur jaringan komunikasi yang diharapkan dapat digunakan pada saat infrastruktur telekomunikasi utama mengalami masalah yaitu menggunakan teknologi *Wireless Sensor Network(WSN)*. Manet dalam hal lain dapat diartikan sebagai node yang terkumpul, lalu bergerak acak(dinamis), kemudian menghasilkan jaringan sementara dengan tidak mengandalkan struktur yang ada. Menggunakan metode *Babel Routing* yang secara otomatis akan mencari node node di sekitarnya. Dengan memanfaatkan Beaglebone Black sebagai node manet, WSN dapat diterapkan sebagai komunikasi antar node. *Beagleboneblack,* ditambah dengan banyaknya pin header ,pin digital, analog, pwm dan lain-lain akan sangat powerfull.Selain menggunakan OS bawaan, kita juga bisa menggunakan OS Linux *Beaglebone*board seperti *Debian 8.6*. Sensor dipasang pada masing masing node, sehingga manet dapat dibangun sebagai jaringan mobile.

Kata kunci : Wireless, Wireless Sensor Network, Manet, Babel Protokol, Beagleboneblack, Debian 8.6

ABSTRACT

Muhammad Rosyid Khulafa, Implementation of Wireless Sensor Networks Using Babel Routing Protocol
Mentor: Sabriansyah Rizqika Akbar, S.T., M.Eng. and Gembong Edhi Setyawan, S.T., M.T

The country of Indonesia has a vast territory requiring defense and a war security strategy is needed to anticipate attacks from various regions. When there is war contact in a certain region, in anticipation of war, as a country that has a long history must have a strong military. It cannot be denied that the occurrence of war always causes damage to the field and command post, besides that the conditions of temperature and weather on the battlefield are very influential on the smooth operation. In addition, the strategy of formation movement in warfare is not only focused on one place, this causes the use of telecommunications infrastructure to not be maximal. In fact, communication in times of war is needed so that the delivery of information can run well. The technological development in the field of wireless communication is Wireless. The one currently developed is the Wireless Sensor Network (WSN). A technology that consists of nodes as spread in the scope of the system using wireless networks. One of the uses of WSN technology is to apply applications to find out the temperature, weather, distance to the surrounding environment, then be able to exchange data through a network of connected node nodes. Over time, the increase in the need for WSN implementation has caused problems. One of them is if there is damage to fixed communications, WSN can be applied as secondary communication or other planning, as a communication network infrastructure that is expected to be used when the main telecommunication infrastructure experiences problems, namely using Wireless Sensor Network (WSN) technology. In other cases Manet can be interpreted as a collected node, then moves randomly, then produces a temporary network without relying on existing structures. Using the Babel Routing method which will automatically search for node nodes around it. By utilizing Beaglebone Black as a manet node, WSN can be applied as communication between nodes. Beagleboneblack, coupled with the number of header pins, digital, analog, pwm, etc. will be very powerful. Besides using the default OS, we can also use Linux Beagleboneboard OS like Debian 8.6. Sensors are installed on each node, so that manet can be built as a mobile network.

Keywords: Wireless, Wireless Sensor Network, Manet, Babel Protocol, Beagleboneblack, Debian 8.6

DAFTAR ISI

PERSETUJUAN 2
PERNYATAAN ORISINALITAS iv
KATA PENGANTAR.....v
ABSTRAK.....vi
ABSTRACT.....vii
DAFTAR ISI.....viii
DAFTAR TABEL xi
DAFTAR GAMBAR.....xii
BAB I PENDAHULUAN .....**Error! Bookmark not defined.**
1.1 Latar Belakang .....**Error! Bookmark not defined.**
1.2 Rumusan Masalah .....**Error! Bookmark not defined.**
1.3 Tujuan 2
1.4 Manfaat 2
1.5Batasan Masalah 3
1.6 Sistematika Pembahasan 3
BAB II LANDASAN KEPUSTAKAAN 5
2.1 Kajian Pustaka.....5
2.2 Mobile Ad-Hoc Network.....5
2.3Wireless Sensor Network 6
2.4 Sensor 6
2.5Topologi Jaringan Ad-Hoc.....7
2.6Protocol Routing pada Manet 7
2.7Protocol Routing Babel.....8
2.7.1Transmisi Pesan dan Penerimaan Pesan 8
2.7.2Proses Route Babel 9
2.7.3Pemeliharaan Tabel Routing.....9
2.7.4Perhitungan metrik.....10
2.7.5Route selection 10
2.7.6Sending Updates 11
2.7.7Packet Format 11
2.7.8Hello Message.....12
2.7.9Message Format 12

2.7.10Router ID 13
2.8Pengiriman Data 13
2.9Program Socket menggunakan TCP di Python 14

BAB III METODOLOGI ..... **Error! Bookmark not defined.**
3.1 Metode Penelitian 16
3.2 Studi Literatur 16
3.3Analisis Kebutuhan 16
3.3.1Analisis Kebutuhan Perangkat Keras 17
3.3.2Analisis Kebutuhan Perangkat Lunak 17
3.4Perancangan 17
3.4.1Perancangan Sistem 18
3.4.2Perancangannode 19
3.4.3OS Debian 8.6 iot ARM 19
3.4.4Protokol Routing Babel 19
3.4.5Aplikasi Pengiriman Data 20
3.5Implementasi 21
3.6Skenario Penelitian 23
3.7Pengujian 23
3.7.1Pengujian Self Configure 23
3.7.2Pengujian Self Healing 24
3.7.3Pengujian Komunikasi Antar Node 24
3.8 Pengambilan Kesimpulan 24
BAB IVREKAYASA KEBUTUHAN 25
4.1 Spesifikasi Lingkungan Sistem 25
4.1.1Spesifikasi Lingkungan Perangkat Keras 25
4.1.2 *Beaglebone* 25
4.1.3SD Card 26
4.1.4*Wifi* 26
4.1.5 Sensor 28
4.2Spesifikasi Linkungan Perangkat lunak 29
BAB VIMPLEMENTASI 30
5.1 Spesifikasi Kebutuhan 30
5.2Instalasi OS Debian 8.6 pada SD Card 30

5.3Setting Wifi 31
5.3.1Instalasi Babel 33
5.3.2Konfigurase Protokol routing Babel 33
5.3.3Pengalamatan Node 33
5.4Rekayasa Kebutuhan 34
5.5Skenario implementasi aplikasi data 35
4.6Kendala dan Solusi 36
BAB VI PENGUJIAN DAN ANALISIS 37
6.1Pengujian dan analisa 37
6.1.1Pengujian *self configure* 37
6.1.2Pengujian *self healing* 40
6.1.3Pengujian letak node manet 41
6.1.4Pengujian Sensor 44
BAB VII PENUTUP 46
7.1Kesimpulan 46
7.2Saran 46
DAFTAR PUSTAKA 47

DAFTAR TABEL

Tabel 3.1Alur Protokol Routing Babel.....20
Tabel 4.1Spesifikasi Beaglebone.....25
Tabel 4.2Spesifikasi SD Card.....26
Tabel 4.3Spesifikasi Wifi.....26
Tabel 4.4Spesifikasi Linkungan Perankat Lunak Komputer.....29

DAFTAR GAMBAR

Gambar 2.1Jaringan Ad-Hoc 7
Gambar 2.2Packet Format 11
Gambar 2.3Pesan Hello.....12
Gambar 2.4Message Format 12
Gambar 2.5Router ID 13
Gambar 2.6Pemrogaman Socket 15
Gambar 3.1Flowchart Metode Penelitian 16
Gambar 3.2Diagram Kebutuhan Sistem 17
Gambar 3.3Pohon Perancangan 18
Gambar 3.4Alur PerancanganSistem Secara Umum 18
Gambar 3.5Beaglebone Black.....19
Gambar 3.6Hubungan Server dengan Beberapa Client.....20
Gambar 3.7Flowchart Aplikasi Pengiriman Data 21
Gambar 3.8Langkah Implementasi Secara Umum 22
Gambar 3.9Rancangan Arsitektur.....23
Gambar 4.1 Beaglebone 26
Gambar 4.2 Micro SD SanDisk 8 GB.....26
Gambar 4.3 Wifi Dongle Edimax EW-7811Un.....27
Gambar 4.4 Sensor LDR 28
Gambar 4.5 Sensor DHT11.....28
Gambar 4.6 Sensor Ultrasonic 29
Gambar 5.1 Implementasi Sistem.....30
Gambar 5.2 Software Win32 Disk Manager 31
Gambar 5.3 Letak Node-node pada Skenario 1.....34
Gambar 5.4 Letak Node-node pada Skenario 2.....34
Gambar 5.5 Letak Node-node pada Skenario 3.....35
Gambar 5.6 Letak Node-node pada WSN.....35
Gambar 6.1 Letak Node-node pada Pengujian Self Configure 38
Gambar 6.2 Node A Tidak Terkoneksi dengan Node C.....38
Gambar 6.3 Node A Terkoneksi dengan Node C lewat Node B.....39
Gambar 6.4 Routing Table Node A Pengujian Self Configure.....39
Gambar 6.5 Pengujian Self Healing.....40

Gambar 6.6 Terjadi Kerusakan Pada Node B 41
Gambar 6.7 Letak Node-node Pada Skenario 1 41
Gambar 6.8 Routing Table Node A pada Skenario 1 41
Gambar 6.9 Routing Ping Node A pada Skenario 1 42
Gambar 6.10 Letak Node-node pada Skenario 2 42
Gambar 6.11 Routing Table Node A pada Skenario 2 43
Gambar 6.12 Ping Node A pada Skenario 2 43
Gambar 6.13 Letak Node-node pada Skenario 3 43
Gambar 6.14 Routing Table Node A pada Skenario 3 44
Gambar 6.15 Ping Node A pada Skenario 3 44
Gambar 6.16 Pengujian Sensor DHT11 44
Gambar 6.17 Pengujian Sensor LDR 45

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Negara Indonesia memiliki wilayah yang luas dengan jumlah penduduk terbesar ketiga di dunia. Pertahanan dan strategi keamanan perang sangat diperlukan untuk mengantisipasi serangan dari berbagai wilayah.Ketika terjadi kontak perang pada wilayah wilayah tertentu, Untuk mengantisipasi peperangan, sebagai negara yang memiliki sejarah panjang haruslah memiliki miiter yang kuat. Tidak dapat dipungkiri bahwa terjadinya perang selalu menyebabkan kerusakan medan dan pos komando,disamping itu kondisi suhu dan cuaca pada medan pertempuran sangat berpengaruh terhadap kelancaran operasi. Selain itu, strategi pergerakan formasi dalam peperangan tidak hanya terfokus pada satu tempat saja,hal ini menyebabkanpenggunaan infrastruktur telekomunikasi tidak dapat maksimal. Padahal, komunikasipada saat terjadi perang sangat diperlukan agar penyampaian informasi dapat berjalan dengan baik.

Salah satu perkembangan teknologi dibidang komunikasi nirkabel adalah *Wireless.* Yan dikembangkan saat ini menjadi *Wireless Sensor Network(*WSN). Sebuah teknologi yang terdiri dari node node sebagaimana tersebar dalam ruang lingkup system menggunakan jaringan nirkabel. Salah satu pemanfaatan teknologi WSN adalah untuk penerapan aplikasi agar mengetahui suhu, cuaca, jarak pada lingkungan sekitarnya, kemudian dapat bertukar data melalui jaringan node node yang telah terhubung.

Seiring berjalannya waktu, meningkatnya kebutuhanyang digunakan dalam penerapan WSNmenimbulkan permasalahan. Salah satunya jika terdapat kerusakan pada komunikasi tetap, WSN dapat diterapkan sebagai komunikasi sekunder atau planning lain, sebagai infrastruktur jaringan komunikasi yang diharapkan dapat digunakan pada saat infrastruktur telekomunikasi utama mengalami masalah yaitu menggunakan teknologi *Mobile ad hoc network* (Manet) yang diterapkan pada *Wireless Sensor Network*(WSN). Manet dalam hal lain dapat diartikan sebagai node yang terkumpul, lalu bergerak acak(dinamis), kemudian menghasilkan jaringan semntara dengan tidak mengandalkan struktur yang ada. Dengan memanfaatkan *Beaglebone Black*sebagai node manet, WSN dapat diterapkan sebagai komunikasi antar node. *Bealebone Black*adalahmini computer produk open source-hardware dengan dukungan linux Angstrom ARM, ditambah dengan banyaknya pin header ,pin digital, analog, pwm dan lain-lain akan sangat powerfull. Selain menggunakan OS bawaan, kita juga bisa menggunakan OS Linux Beagleboard ubuntu(Saucy, 2014). Sensor yang dipasang pada masing masing node, manet dapat dibangun sebagai jaringan mobile.

Setiap node dalam *Manet*merupakan node bebas, bergerak secara acak dan berpindah-pindah sesuai ruang lingkupnya, hal ini menyebabkan topologi jaringan *nirkabel* dapat berubah cepat dan tak terduga. Dengan memanfaatkan antar muka *nirkabel* untuk menghubungkan antara satu node dengan node lainnya, untuk mengimplementasikan *Manet* pada daerah perang, dibutuhkan aplikasi pengiriman pesan untuk sistem komunikasi darurat pada jaringan *Manet* selanjutnya pengiriman pesan tersebut dapat berupa data, dengan memanfaatkan

sensor pendeteksi jarak, suhu, kelembapan dan cahaya sebagai langkahimplementasi *wireless sensor network.*

1.2 Rumusan Masalah

Berdasarkan permasalahan dari latar belakang tersebut, dapat dirumuskan masalah sebagai berikut:

1. Bagaimana rancangan komunikasi *pengiriman data* dengan memanfaatkan *Manet*?
2. Bagaimana alur dan pengaruh *node* terhadap perubahan topologi yang berubah ubah?
3. Bagaimana program *wireless sensor network*bertukar data melalui infrastruktur *Manet?*

1.3 Tujuan

Tujuan:

1. Melakukan uji coba sistem komunikasi menggunakan aplikasi *pengiriman data* pada *Manet*dengan menggunakan *Beaglebone Black* untuk komunikasi jaringan yang mempunyai topologi dinamis.
2. Mendapatkan hasil pengukuran sensor pada tiap tiap node.

1.4 Manfaat

Dengan Memanfaatkan Teknologi *Manet* dan penerapan Wireless Sensor Network menggunakan *Beaglebone Black*untuk sistem komunikasi efektif pada area perang. Sebagaimana agar memenuhi persyaratan akademik dan administratif Universitas, penulis diharapkan mampu menerapkan dan mendapatkanilmu tambahan,sehingga berguna untuk kedepannya dan penulis sendiri. Adapaun manfaat yang diharapkan adalah

- Bagi penulis
  1. Media penyalurandanmenerapkanilmu yang didapatdan teknologi yang semakin berkembang pesat,salah satunya di bidang perencangan jaringan, Sensor jaringan dan komunikasi jaringan.
  2. Mendapatkan pengetahuan lebih dan wawasan pengembangan teknologi Manet berbasis *Wireless Sensor Network* untuk komunikasi.
- Bagi pembaca
  1. Mendapatkan wawasan tentang teknologi Manet berbasis Wireless Sensor Network.
  2. Mendapatkan wawasan terkait sistem komunikasi jaringan menggunakan pemrogaman *socket*.
  3. Dengan adanya penerapan teknologi Manet ini, dapat memudahkan proses komunikasi ketika terjadi suatu peperangan.
  4. Pembaca yang memiliki minat dibidang Wireless Sensor Network dapat mengembangkan pengetahuan lebih yang nantinya dapat digunakan sesuai kebutuhan.

1.5 Batasan Masalah

Setiap permasalahan yang diteliti, selayaknya memiliki Batasan Batasan terstruktur, dilakukan agar masalah yan kita bahas tidak terlalu jauh dari topik. Batasan Batasan tersebut antara lain:

1. Operasi *Debian 8.6*yang khusus digunakan untuk hardware *Beaglebone Black.*
2. Menggunakan jaringan *WirelessEdimax.*
3. Menggunakan sensor *DHT11, Ultrasonic dan LDR* pada masing masing Beaglebone.
4. Menggunakan tiga node *Beaglebone Black*sebagai simulasiperpindahan*node.*
5. Penelitian dalam hal ini tidak meninjau aspek keamanan pada jaringan.
6. Aplikasi pengiriman datayang digunakan adalah bahasa *Python.*
7. Penelitian ini juga mengabaikan aspek jarak antar *mobile node* yang terlalu jauh.
8. Penelitian juga mengabaikan kualitas dan akurasi sensor yang digunakan.

1.6 Sistematika Pembahasan

BAB I : **Pendahuluan**

Sebuah pedoman dan alasan mengapa penulisa membahas topik penelitian. Isinya tentanglatar belakang, rumusan masalah, tujuan, manfaat, batasan masalah kemudian sistematika pembahasan.

BAB II **: Landasan Kepustakaan**

Menjelaskan tentang kajian pustaka yang diambil, teori yang memperkuat perkembangan teknologi *manet* pada sistem komunikasi *Pengiriman data*menggunakan bahasa *Python.*

BAB III **: Metodologi**

Dalam bab ini akan menjelaskan mengenai metode kemudian langkah kerja yang dilakukan penulis, terdiri atas studi literatur, uraian tentang kebutuhan perangkat lunak dan perangkat keras, perancangan node, implementasi, analisis serta pengujian terakhir kesimpulan.

BAB IV **: Rekayasa Kebutuhan**

Bab ini membahas mengenai tahapan rekayasa *manet* dan pengiriman data yang diambil dari proses analisis kebutuhan dan perancangan perangkat.Mulai dari spesifikasi sistem, konfigurasi jaringan dan tahapan implementasi.

BAB V **: Implementasi**

Bab ini menjelaskan mengenai tahapan tahapan pada perencanaan implementasi dalam Wireless sensor network menggunakan babel routing protocol.

BAB VI : Pengujian dan Analisis

Membahas mengenai uji coba dan analisa sistem komunikasi pada *Manet*. Proses pengujian dilakukan melalui 2 macam pengujian yakni pengujian *Manet* dan pengujian pengiriman data sensor. *Manet* digunakan agar memperoleh hasil apakah tiap *node* bisa berkoneksi secara otomatis menggunakan protokol *routingbabel*, pada pengujian *Manet* ini dibagi lagi menjadi dua pengujian yaitu *self configure* dan*self healing*. Pengujian pengiriman data sensor dilakukan setelah node dalam manet terhubung satu sama lain.Dari hasil pengujian tersebut, diharapkan penelitian ini bisa diterapkan di daerah bencana yang sebenarnya.

BAB VII : Penutup

Pada bagian ini, hasil dari penelitian pada bagian bagian sebelumnya akan dianalisa kemudian mengambil kesimpulan. Kemudian saran bagi pembaca agar melakukan pengembangan teknologi selanjutnya.

BAB 1 LANDASAN KEPUSTAKAAN

1.1 Kajian Pustaka

Dari penelitian yang telah dilakukan peneliti peneliti seblumnya, dapat dijadikan tolak ukur agar menjadikan referensi terkait *Wireless Sensor Network* dan *Manet.* Berikut kajian terhadap beberapa hasil penelitian:

Pada tahun 2007 Tofan Teguh Perkasa dengan penelitian berjudul Analisis Kinerja *Voice Over Internet Protocol pada Manet*.Menghasilkan gambaran tentang Voip layak dan berhasil dilewatkan pada *Manet.*

Pada tahun 2008 Tri Hadiah Muliawati dengan penelitian berjudul *Pengiriman Pesan Secara Berantai Pada Daerah Perang Terisolasi Menggunakan Teknologi Manet. Memanfaatkan teknologi Bluetooth pada handphone untuk pengiriman pesan berantai menggunakan teknologi manet.*

Pada tahun 2012 I Komang Ari Mogi dengan penelitian yang berjudul *Vanet Untuk Solusi Komunikasi Data di Kawasan Pariwisata.* Diterapkan di Kawasan wisata Bali karena tidak dapat dibangun nya infrastruktur komunikasi akibat hokum adat yang membatasi pembangunan daerah di Kawasan pariwisata.

Pada tahun 2013 I Nyoman Buda Hartawan dan Waskitho Wibisono dengan penelitian berjudul Mekanisme Pemilihan MPR dengan Congestion Detection dalam OLSR pada Manet.

Dari beberapa contoh hasil peneliti diatas, maka dapat digambarkan beberapa persamaan perbedaanya. Persamaan penelitian ini dengan hasil-hasil penelitian sebelumnya adalah pada salah variabel yang digunakan dalam membahas pokok permasalahan, yaitu variable *Manet* dan komunikasi.

Sedangkan, perbedaan antara penelitian ini dengan hasil-hasil penelitian yang ada adalah pada kaitan pembahasan variabel *Manet* itu sendiri. Pada penelitian ini peneliti menggunakan *routing* protokol *Babel* untuk membangun jaringan *Manet* dan untuk komunikasinya menggunakan*Wireless Sensor Network*. Sementara itu, pada penelitian disebutkan diatas menjelaskan *Manet* dengan menggunakan perangkat dan protokol yang berbeda dengan penelitian ini.

1.2 Mobile Ad-Hoc Network

Mobile ad-hoc network(manet), node pada manetyang berbeda dapat terhubung melalui wireless secara langsung, dalam hal ini node satu dengan node lain haruslah dalam jangkauan. Jika salah satu node diluar jangakauan, maka dibutuhkan node lain untuk meneruskan pesan. Oleh karena itu muncullahsimulasi multi hop, yang mana ada beberapa host berfungsi sebagai relay,yang digunakan untuk meneruskan paket dari host/server menuju kepada target/client.(Binus, 2011)

Manet menjadi subjek yang sangat popular untuk penelitian karena adanya laptop dan teknologi *Wireless*yang banyak digunakan pada pertengahan akhir tahun 1990-an. Penerapan

Manet terutama di bidang militer, pada situasi darurat (perang) yang tidak didukung oleh infrastruktur komunikasi dan penggunaan seperti *sharing file* atau perangkat keras antar beberapa *node* di rumah maupun di perkantoran. Jaringan ini dipilih karena penerapannya yang cenderung lebih murah, cepat, mudah dikonfigurasi dibandingkan jaringan infrastruktur *Wireless* maupun jaringan kabel. Dalam *Manet*, setiap *node*pada manet tidak hanya sebagai penerima maupun pengirim, tetapi juga dapat meneruskan pesan antar *node* di sekitarnya sebagai router. Untuk itu dibutuhkan *routing protocol* untuk membantu tiap-tiap *node* melakukannya. *Routingprotocol* untuk *Manet*sangat berbeda dengan *routingprotocol*yang biasanyadigunakan pada infrastruktur tetap atau jaringan kabel. Karena pada dasarnya pergerakan topologi*manet* yang dinamis dan cenderung bergerak berubah ubah, berbeda halnya dengan jaringan kabel yang cenderung tetap.(Binus, 2011)

1.3 *WirelessSensor Network*

Teknologi *Wireless*dapat diartikan teknologi tanpa kabel. *Wireless sensor network* terdapat komponen dua, sensor dan mikrokontroller. Sensor melekat pada mikrokontroller. Mikrokontroller diberikan tugas untuk penginderaan fenomena fisik seperti suhu cuaca kelembapan maupun jarak.Pada tahap selanjutnya yakni melanjutkan informasi yangadadari penerapan pengolahan data, kakrena ituada komponen utama lainnyaseperti protocol komunikasi yang disetting untuk mengatur langkah langkah interaksi sensor node.(C.S Raghavendra, 2006)

1.4 Sensor

Sensor yakni sejenis tranduser, biasa digunakan untuk mengubah suatu besaran. Magnetis, panas, sinar, kimia, mekanis, kemudian dirubah menjadi arus listrik maupun tegangan. Sensor biasa digunakan untuk mendeteksi saat uji coba pengukuran maupun pengendalian.

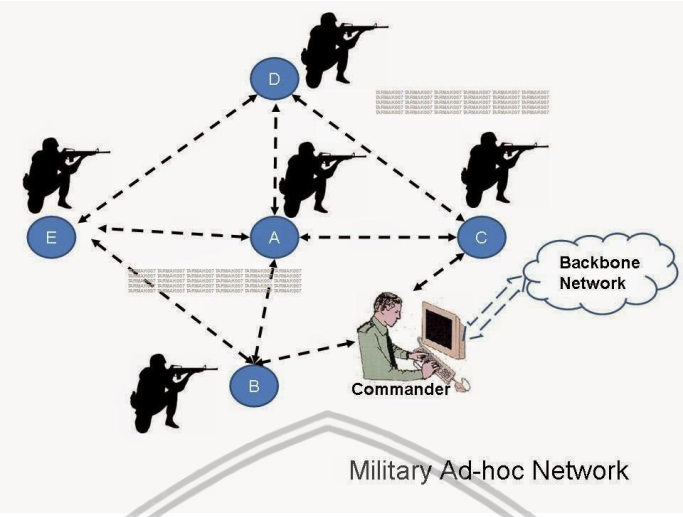
1. DHT11: DHT11 merupakan sensor untuk mensensing suatu ruangan, yakni suhu dan kelembaban dalamsatu alat, dimana hasil sinyal digital yang yang sudah di tentukan batasannya. Memiliki kualitas data sensing yang cepat.
2. Ultrasonic: Merupakan sensor jarak yang mana diukur suatu jarak dengan mengirim gelombang suara dalam jarak tertentu, kemudian gelombang tersebut dipantulkan kembali. Waktu antara gelombang dipantulkan dan dikembalikan merupakan jarak yang diitempuh sensor dan objek.
3. Light Dependent Resistor (LDR): Mengubah cahaya dengan infra merah dari spektrum menjadi sinyal listrik. DIkenal sebagai fotoelektrik dari foton menjadi electron.

1.5 Topologi Jaringan Ad-Hoc

Alur topologi jaringan ad-hoc yang terhubung dari dua atau lebih node salin berkomunikasi tanpa menggunakan *akses*. Wireless pada setiap node

N*ode* pada setiap konfigurasi *ad-hoc* memiliki cakupan pada area tertentu. Setiap node yang disimulasikan dapat bertukar data atau berkomunikasi jika masih dalam ruang lingkup atau area

tertentu. Karena itu konfigurasi ad-hoc dapat berperan sebagai router maupun host untuk mengarahkan data menuju tujuan. (Atul Chaturvedi, 2015)



Gambar 2.1 Jaringan Ad-Hoc

Sumber :(Atul Chaturvedi, 2015)

1.6 Protocol *Routing* pada *Manet*

Terdiri dari dua macam protocol, sangat menarik bagi pengembangan jaringan nirkabelsetelah perkembangan node yang dapat bergerak bebas:

1. Proaktif: Node yang terhubung menyimpan table(table driven), dengan kata lain informasi informasi node yang diketahui tersimpan di dalamnya.

   Contoh: *Babel, Wireless Routing protocol(WRP),Optimized Linkstate (OLSR). Destination Sequenced Distance Vector (DSDV), Cluster Switch Gateway Routing (CSGR).*

2. Reaktif: Hanya membentuk node yang berasal dari node asal menuju node yang dituju, ini berdasar pada sumber node awal.

   Contoh: *Ad-hoc On-demand Distance Vector (AODV), Temporally Ordered Routing Algorithm (TORA), Dynamic Source Routing (DSR), Associativy Based Routing (ABR), Signal Stability Routing (SSR).*

Hal lain menerangkan uji coba pada *protocol routing hybrid* yang menggabungkan antara kedua tipe protokol *routing*, reaktif dan proaktif, contohnya *Zone Routing Protocol (ZRP).*(Staub, 2004).

1.7 Protokol Routing Babel

Babel merupakan protokol routing distance-vector loop-avoiding untuk IPv6 dan IPv4 dengan properti konvergensi yang cepat. Hal ini didasarkan pada gagasan di

DestinationSequenced DistanceVector Routing(DSDV), Ad-hoc On Demand Distance Vector(AODV) dan Cisco,namun dirancang untuk bekerja dengan baik tidak hanya di jaringan kabel tetapi juga di jaringan mesh nirkabel, dan telah diperpanjang dengan dukungan untuk jaringan overlay. Babel sedang dalam proses menjadi Standar IETF.

Pada jaringan dual stack, babel dapat menggunakan IPv4 dan IPv6 secara bersamaan.

1. Kehandalan*babel* yakni komunikasi menjamin tidak terjadinya routing loop, jika update rute tidak memenuhi kondisi. Kelebihannya, ketika rute metric pada local node tidak lebih besar dari metric rute yang saat ini dipilih.(J. Chroboczek, 2011)

Babel memiliki ketentuan untuk kualitas estimasi link dan metrik yang cukup sewenang-wenang. Ketika dikonfigurasi sesuai, *Babel* dapat menerapkan *pathrouting* terpendek, atau mungkin menggunakan berbasis *metrik* pada statistik *packet* loss. *NodeBabel* akan berhasil mendirikan sebuah asosiasi bahkan ketika mereka dikonfigurasi dengan parameter yang berbeda. Sebagai contoh, sebuah *mobilenode* yang rendah pada baterai dapat memilih untuk menggunakan konstanta waktu yang lebih besar (*Hellomessage* dan interval pembaruan, dll) dari sebuah *node* yang memiliki akses ke listrik di dinding. Sebaliknya, simpul yang mendeteksi tingkat *mobilitas* yang tinggi dapat memilih untuk menggunakan konstanta waktu yang lebih kecil. Kemampuan untuk membangun jaringan heterogen seperti membuat *Babel* khususnya disesuaikan dengan lingkungan *nirkabel*. *Babel* adalah protokol *routing* hybrid, dalam arti bahwa hal itu dapat membawa *rute* untuk beberapa protokol lapisan jaringan (*IPv4* dan *IPv6*). (J Chroboczek, 2011).

1.7.1 *Transmisi* Pesan dan Penerimaan Pesan

Pertukaran pesan pada *Babel*, satu atau lebih pesan *Babel* yang ditambahkan untuk membentuk paket *Babel*, yang dikirim dalam *datagram UDP* tunggal. Alamat sumber dari paket *Babel* selalu alamat *unicast link-local*. Paket *Babel* dapat dikirimkan ke alamat *multicast link-local* atau ke (*link-local*) alamat *unicast*. Dalam operasi normal, seorang pembicara *Babel* mengirimkan paket *multicast* dan *unicast* baik dengan tetangganya. Dengan pengecualian pesan *Hello* dan *acknowledgment*, semua pesan *Babel* dapat dikirim ke alamat baik *unicast* atau *multicast*, dan semantik mereka tidak tergantung pada apakah tujuan adalah alamat *unicast* atau *multicast*. Oleh karena itu, seorang pembicara *Babel* tidak perlu menentukan alamat tujuan dari sebuah paket yang diterima untuk menafsirkannya.

Sebuah jumlah *jitter* diterapkan pada pesan yang dikirim oleh *Babel*, keluar pesan yang *buffered*, dan harus dikirim dengan penundaan acak kecil. Hal ini dilakukan untuk dua tujuan menghindari sinkronisasi beberapa pembicara *Babel* melalui jaringan (*jitter*), dan memungkinkan untuk *agregasi* beberapa pesan ke dalam satu paket. *Delay* yang tepat dan jumlah *jitter* diterapkan pada pesan tergantung pada apakah pesan sangat mendesak atau tidak. Pesan *acknowledment* harus dikirim sebelum batas waktu yang ditentukan dalam permintaan yang sesuai. Kelas tertentu pesan pembaruan yang ditentukan harus dikirim pada waktu yang tepat. Kelas permintaan tertentu dan memperbarui pesan harus dikirim pada waktu yang tepat. (J Chroboczek,2011).

1.7.2 Proses *Route Babel*

Tabel *rute* di indeks tiga kali lipat dari bentuk (*prefix, plen, neighbour*), dan setiap entri tabel *rute* berisi *data* sebagai berikut:

1. *Advertised prefix(prefix, plen).*
2. *Neighbour* yang di *advertised* pada *rute* ini.
3. *Metrik* dengan *rute* ini di *advertised* oleh tetangga, yang dikenal sebagai referensi *metrik rute* itu, atau *0xFFFF* (tak terhingga) untuk *rute* yang baru ditarik.
4. Nomor urutan dengan *rute* yang telah di *advertised*.
5. alamat hop berikutnya dari *rute* ini.
6. Sebuah *flag* yang menunjukkan apakah *rute* ini dipilih, yaitu apakah saat ini sedang digunakan untuk meneruskan dan yang di *advertised*.
7. Ada satu waktu terkait dengan setiap entri tabel *rute*, *rute* berakhirnya *timer*. Hal ini dijalankan dan *reset*. (J Chroboczek, 2011).

1.7.3 Pemeliharaan Tabel *Routing*

Secara konseptual, *update Babel* adalah *quintuple* (*prefix, plen, Router-Id , seqno, metric*), di mana *(prefix, plen)* adalah *prefix* untuk *rute* yang *advertised*, *Router-Id* adalah nomor *router* yang berasal dari *updateroute*, *seqno* adalah nomor *advertised* yang berurutan, tanpa pengurangan *(modulo 2 ^ 16)integer* yang didefinisikan oleh *router* berasal, dan mengumumkan *metrik*. Sebelum diterima, pembaruan tersebut diperiksa terhadap kondisi kelayakan, suatu kondisi yang memastikan bahwa *rute* tidak membuat *routingloop (dual)*. Jika kondisi kelayakan tidak benar, pembaruan diabaikan atau diperlakukan sebagai pencabutan, tergantung pada beberapa kondisi lain. Jika kondisi kelayakan benar, maka *update* tidak mungkin menyebabkan *routingloop* , dan *update* diterima. Sebelum *advertisingrute*, *update* tabel *nodeBabel* sumber dengan informasi yang akan diperlukan untuk mengevaluasi kondisi yang terjadi. (J Chroboczek, 2011)

1.7.4 Perhitungan Metrik

Metrik Sebuah *rute* yang dihitung dari referensi metrik-metrik tetangga yang di *advertised*, dan biaya link penyebaran tetangga. Sama seperti perhitungan link, perhitungan metrik dianggap sebagai masalah kebijakan local, sejauh *Babel* yang bersangkutan, fungsi M (c, m) yang digunakan untuk komputasi metrik dari biaya neighbour dan referensi *rute* ini metrik harus hanya memenuhi persyaratan sebagai berikut:

1. jika c adalah tidak terbatas, maka M (c, m) tidak terbatas.
2. M adalah tidak berubah: M (c, m)> m.
3. Metrik harus memenuhi kondisi berikut:
   M adalah *isotonik*: jika m <= m' maka M (c, m) <= M (c, m').

Perhatikan bahwa sementara *monotonicity* yang ketat sangat penting untuk integritas jaringan (*loop routing* yang persisten dapat muncul jika tidak benar), isotonisitas tidak, jika tidak benar, *Babel* masih akan konvergen ke tabel *routing* lokal secara optimal, tetapi tidak berarti

mencapai optimum *global* (pada kenyataannya, seperti *optimum global* bahkan mungkin tidak ada. (J Chroboczek - 2011).

1.7.5 Route Selection

Pemilihan *rute* adalah proses dimana satu *rute* untuk awalan yang diberikan dipilih untuk digunakan untuk paket forwarding dan akan *readvertised* ke tetangga sebuah *node*. *Babel* ini dirancang untuk memungkinkan kebijakan pemilihan *rute* yang fleksibel. Sejauh kebenaran protokol yang bersangkutan, kebijakan pemilihan *rute* hanya harus memenuhi sifat-sifat sebagai berikut:

1. *rute* dengan metrik yang tak terbatas tidak pernah dipilih.
2. *rute* yang tidak layak tidak pernah dipilih.

Bagaimanapun, *Babel* tidak secara alami menjamin stabilitas *routing*, dan mengkonfigurasi kebijakan pemilihan *rute* bertentangan pada *router* yang berbeda dapat menyebabkan persisten *osilasirute*. Mendefinisikan kebijakan pemilihan *rute* yang baik untuk *Babel* merupakan masalah penelitian yang masih terbuka. Pemilihan *rute* dapat mempertimbangkan beberapa kriteria yang saling bertentangan, yaitu:

1. *rute* dengan metrik kecil sebaiknya diutamakan dibanding *rute* dengan metrik besar.
2. perubahan *Router-Id* harus dihindari.
3. *rute* melalui tetangga stabil sebaiknya diutamakan dibanding *rute* melalui yang tidak stabil.
4. *rute* yang stabil sebaiknya diutamakan dibanding yang tidak stabil.
5. beralih *hop* berikutnya harus dihindari.

Sebuah strategi sederhana adalah untuk memilih *rute* layak dengan metrik terkecil, dengan sejumlah kecil *hysteresis* diterapkan untuk menghindari beralih *Router-Id*. Setelah prosedur pemilihan *rute* dijalankan, *update* dipicu dan permintaan dikirim. (J Chroboczek, 2011)

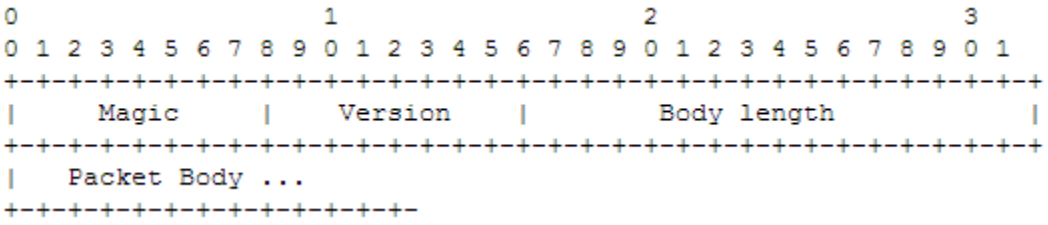
1.7.6 Sending Updates

Babel menyebarkan dengan tetangganya set dari *rute* yang dipilih. Biasanya, hal ini dilakukan dengan mengirimkan satu atau lebih paket *multicast* berisi pesan *Update* pada semua *interface* yang terhubung, namun pada teknologi *multicast link* di mana secara signifikan lebih mahal daripada *unicast*, *node* mungkin memilih untuk mengirim beberapa salinan pembaruan dalam paket *unicast* ketika jumlah tetangga kecil. Selain itu, dalam rangka untuk memastikan bahwa setiap *blackhole* yang andal dibersihkan secara tepat waktu, *nodeBabel* mengirimkan retraksi (*update* dengan metrik tak terbatas) untuk setiap prefiks baru ditarik.

Jika *update* untuk *rute* disuntikkan ke dalam domain *Babel* oleh *node* lokal (misalnya alamat antarmuka lokal, awalan jaringan langsung terpasang, atau menyebarkan dari protokol *routing* yang berbeda), *Router-Id* diatur ke *id* lokal, metrik diatur untuk beberapa nilai terbatas sewenang-wenang (biasanya 0), dan *seqno* diatur ke nomor urut *router* lokal. Jika pembaruan untuk *rute* pembelajaran dari *Babel* speaker yang lain, *Router-Id* dan nomor urut yang disalin dari entri tabel *routing*, dan metrik dihitung. (J Chroboczek, 2011)

1.7.7 Packet Format

Sebuah paket *header Babel* terdiri dari empat oktet, diikuti oleh urutan pesan*Babel.*



Gambar 2.2*packet format*

Sumber: (J Chroboczek, 2011)

Fields :

Magic : Berubah-ubah tetapi hati-hati memilih nilai empat puluh dua(decimal), paket dengan nilai pertama berbeda dari empat puluh dua harus dihiraukan.

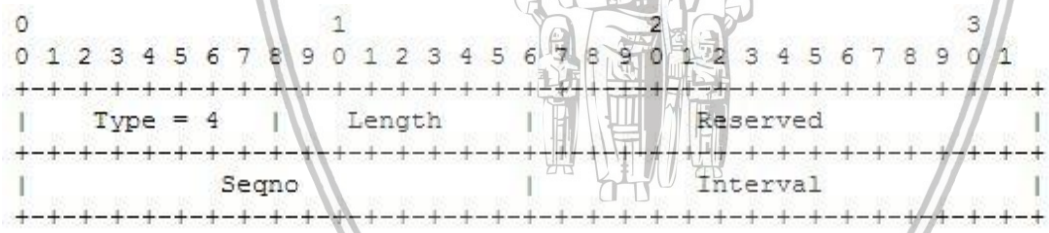
Version *:* Dokumen ini menetapkan versi 2 dari protokol *Babel*. Paket dengan oktet kedua berbeda dari 2 harusdiabaikan.

BodyLength *:* Panjang dalam header octet mengikuti header paket(TLV).

Body : *bodypacket* adalah sebuah urutan pesan.

Setiap *data* yang mengikuti *body* harus diabaikan. (J Chroboczek, 2011)

1.7.8 Hello Messege



Gambar 2.3 Pesan *Hello*

Sumber: (J Choboczek-2011)

TLV ini digunakan untuk penemuan tetangga dan untuk menentukan biaya metrik link ini.

Fields :

Type Di set 4 untuk mengindikasikan sebuah pesan *Hello* TLV.

Length Panjang *body*, jenis ekslusive dan panjang *field*.

Reserved *Field* ini dikirim sebagai 0, dan harus diabaikan pada penerimaan.

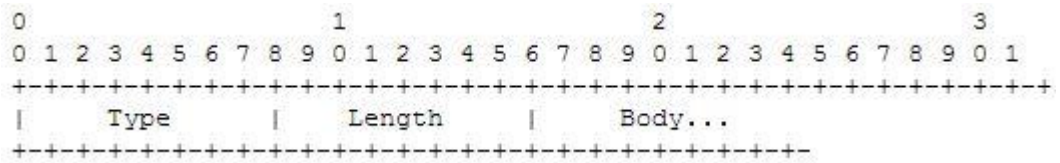
Seqno Nilai dari pengirim *node* hallo seqno untuk *interface* ini.

Interval Batas atas, dinyatakan dalam centiseconds, pada waktu setelah *node* pengirim akan mengirim *Hello* TLV baru. Ini tidak harus 0.

Karena ada seqno tunggal yang membalas untuk semua *Hello* dikirim oleh *node* yang diberikan melalui antarmuka yang diberikan, TLV ini HARUS dikirim ke tujuan multicast. Untuk menghindari diskontinuitas besar dalam kualitas link, beberapa TLV *Hello* tidak harus akan dikirim dalam paket yang sama. (J Chroboczek, 2011)

1.7.9 Messege Format

Dengan pengecualian pada *Padl,* semua pesan harus mengikuti struktur berikut:



Gambar 2.4*Message Format*

Sumber: (J Chroboczek, 2011)

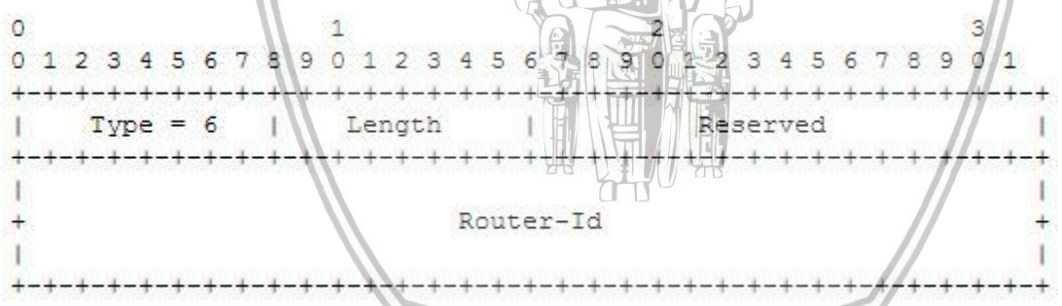
Fields:

Type Kolom ini menentukan jenis pesan.

Length Panjang *body*, *type* eksklusif dan panjang *field*. Jika *body* lebih panjang dari panjang yang diharapkan dari suatu jenis pesan, *data* tambahan harus diam-diam diabaikan.

Body *Body* pesan, penafsiran yang tergantung pada jenis pesan. Jenis pesan yang tidak diketahui harus diabaikan. (J Chroboczek, 2011)

1.7.10 Router ID



Gambar 2.5 Router ID

Sumber: (J Chroboczek, 2011)

Sebuah pesan Router-Id menetapkan Router-Id yang tersirat oleh update pesan berikutnya.

Fields:

Type Mengatur sampai 6 untuk menunjukkan pesan Router-Id .

Length Panjang body, eksklusif type dan Panjang field.

Reserved Bidang ini dikirim sebagai 0, dan harus diabaikan pada penerimaan.

Router-Id Field ini berisi Router-Id untuk rute disebarkan di dalam update pesan berikutnya. (J Chroboczek,2011)

1.8 Pengiriman data

Pengiriman data dapat diartikan komunikasi antar arah. Dalam teknologi komunikasi, pengiriman data diartikan berkomunikasi menggunakan alat komunikasi, bisa berupa teks, suara, atau gambar. Pada jaringan internet, pengiriman data digunakan untuk berbicara dengan manusia melewati komunikasi nirkabel.

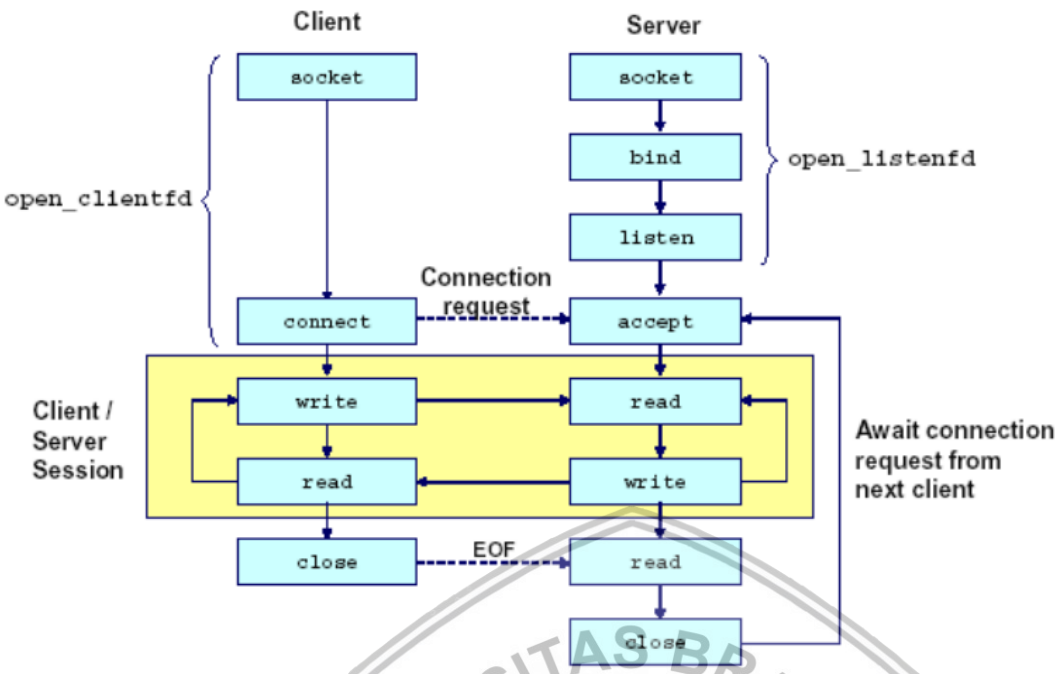
Pengiriman data terbagi menjadi dua jenis, yakni grup(umum) dan privat(khusus). Pengiriman data grup memungkinkan komunikasi lebih dari satu orang, sedangkan khusus, komunikasi hanya tertuju pada satu arah dan tidak diketahui orang lain.(Muhammad Fauzi Haryadi, 2010)

1.9 Program*Socket* menggunakan TCP di *Python*

Dalam sebuah aplikasi jaringan, dua alat yang digunakan menjadi sebuah node, dapat berkomunikasi keduanya melalui socket. Socket merupakan sebuah pintu yang mana proses tersebut harus masuk melewati pintu. Pada saat proses three way handshaking, terjadi connection establishment, dimana client akan mengecek apakah port telahtersedia dan terbuka. ketika port tersedia, maka server akan membuat socket baru(connection socket) untuk melayani client.Pada tahap data exchange yakni ketika koneksi telah terbentuk, terjadi pertukaran data(enkapsulasi).Pada tahap connection termination yakni ketika pertukaran data selesai, terdapat tahap permintaan untuk memutuskan koneksi.(Hero.Lecture, 2011)

Langkah-langkah ketika membuat sebuah program dengan *socket* ditunjukkan pada gambar 2.7, langkah-langkahnya adalah sebagai berikut:

1. Server membuat welcoming socket pada port yang selalu menunggu setiap permintaan untuk diberikan pada client
2. Membuat koneksi client ke server, tahap awal membuat socket dengan melakukan pengalamatan ke server lalu menghubungkan client dan server berupa TCP connection.
3. Client melakukan komunikasi pengiriman dan penerimaan data menggunakan read dan write.
4. Client menutup hubungan dengan perintah close
5. Server menyambungkan hubungan dengan client dengan perintah bind
6. server mempersiapkan socket untuk menerima koneksi dari client yang masuk
7. server melakukan komunikasi pengiriman dan penerimaan data menggunakan read dan write
8. Server menutup hubungan dengan perintah close.

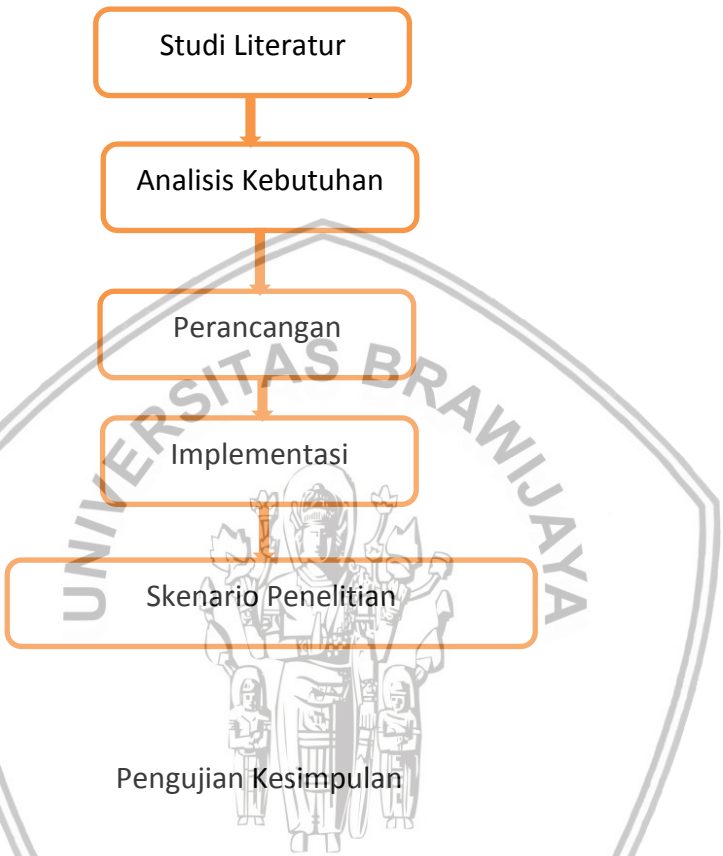


Gambar 2.6 pemrograman *socket*
Sumber: (Hero.Lecturer, 2011)

BAB 1 METODOLOGI PENELITIAN

1.1 Metode Penelitian

Bab inimenerangkan langkah-langkah yang akan diterapkanpada penyusunan skripsi, yaitu skenario penelitian meliputi metode, subjek, lokasi, Teknik pengumpulan data Teknik analisis dan pembahasan hasil, peralatan pendukung yang digunakan. Diagram gambaran dapat dilihat pada 3.1.



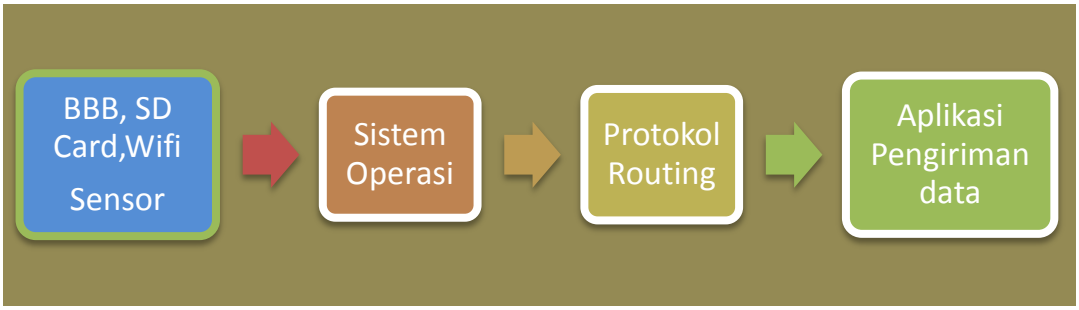
Gambar 3.1 *Flowchart* Metode Penelitian

1.2 Studi Literatur

Tujuan dari Studi literatur untuk mempelajari dan memahami konsep yang terkait dengan IMPLEMENTASI WIRELESS SENSOR NETWORK MENGGUNAKAN BABEL PROTOKOL.Langkah langkah yang sebelumnya menjadi referensi maupun teori teori akan digunakan sebagai kerangka, yang nantinya dapat menunjang pengerjaan skripsi.

1.3 Analisis Kebutuhan

Ditujukan untuk merancang alur pada proses implementasi yang akan dilakukan. Seperti Kebutuhan perangkat keras maupun perangkat lunak.



Gambar 3.2 Diagram kebutuhan sistem

Kebutuhan sistem yang diperlukan dalam penelitian ini adalah *Beaglebone Black, SD card* untuk storage system, dan *wifi* sebagai perangkat nirkabel untuk media komunikasi menggunakan *signal radio*, sistem operasi yang digunakan *operating systemdebian 8.6 iotARM* yang digunakan untuk *Beaglebone Black*, di dalam sistem operasi *debian 8.6* ini protokol *routingbabel* di *install* untuk membangun infrastruktur *Manet*. Aplikasi *Pengiriman data*digunakan untuk komunikasi pengiriman pesan berbasis teks.

1.3.1 Analisis Kebutuhan Perangkat Keras

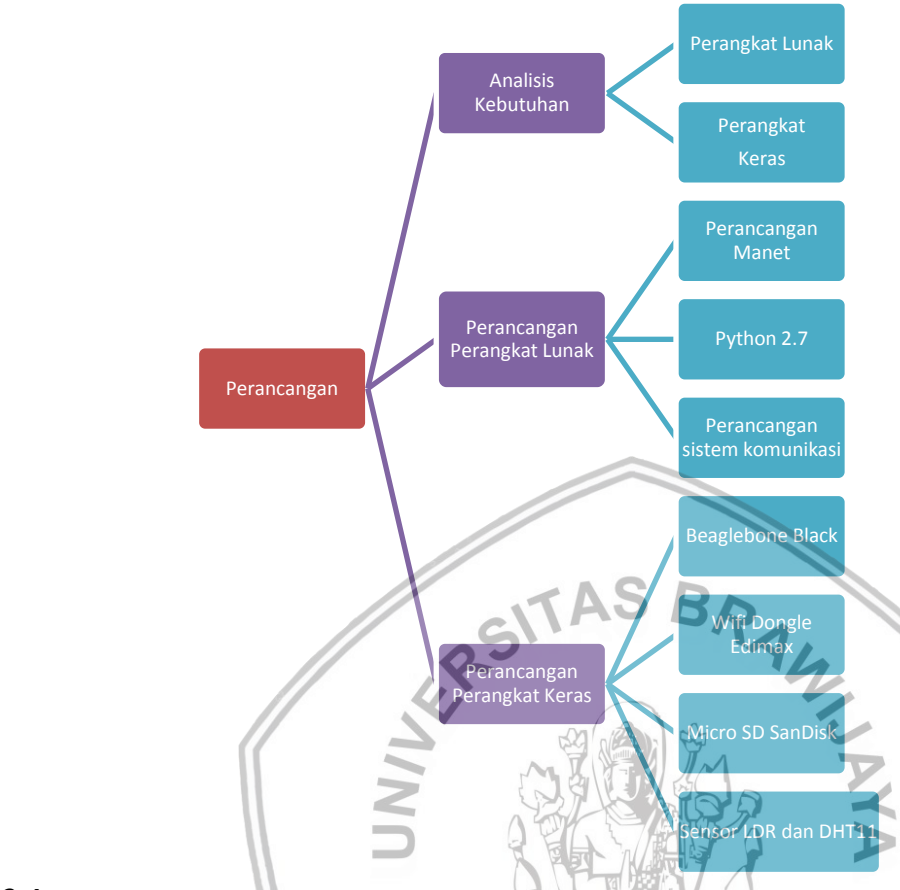
a. *Beaglebone Black* digunakan sebagai *nodeManet*, yang terinstal protokol *routingbabel* dan aplikasi *Pengiriman data*.
b. *Wifi edimax EW-7811Un* untuk menghubungkan dengan perangkat *nirkabel* lainnya atau *nodeManet* lain.
c. *Micro SD SanDisk 8 GB* sebagai tempat penyimpanan *data* dan *operating system Debian* pada *Beaglebone Black*.
d. *Sensor*DHT11 dan LDR digunakan untuk mengetahui kondisi lingkungan dan komunikasi pengiriman data antar *nodeManet*.

1.3.2 Analisis Kebutuhan Perangkat Lunak

a. Protokol *routing babel* digunakan untuk membuat jaringan *Manet.*
b. *Python 2.7* untuk *compiler* aplikasi *Pengiriman data*.
c. Aplikasi *Pengiriman data* untuk sistem komunikasi antar *node* dengan menggunakan *SPI* lewat terminal *linux.*
d. *PuTTy* untuk mengetahui koneksi antar route.

1.4 Perancangan

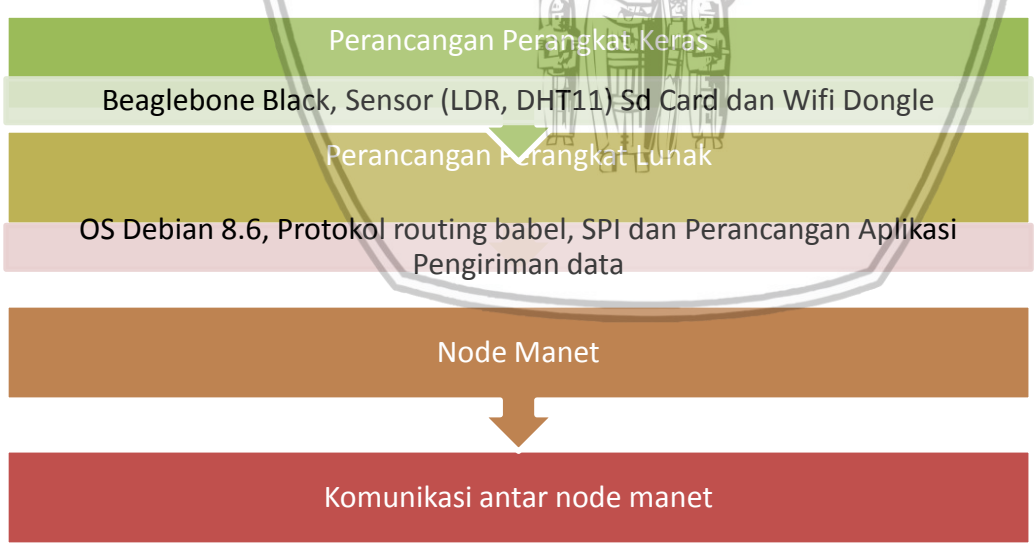
Perancangan jaringan menggunakan *protokol routing babel.* Dalam hal ini dijelaskan pada sub bab 3.4 yakni perancangan perangkat keras maupun lunak, Gambaran umum perancangan akan dijelaskan dalam gambar pohon di bawah ini.



Gambar 3.3 Pohon perancangan

1.4.1 Perancangan Sistem

Dibutuhkan dilakukan agar mengetahui spesifikasi kebutuhan secara menyeluruh. Seperti yang ditunjukkan dalam gambar 3.4.



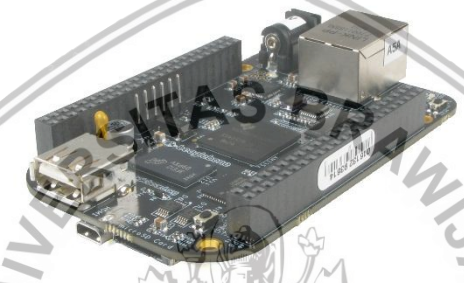
Gambar 3.4 Alur perancangan sistem secara umum

Pada gambar 3.3 dijelaskan mengenai perancangan sistem secara keseluruhan, dimulai dari merancang perangkat keras sistem, yang terdiri dari *Beaglebone Black, Sensor LDR, DHT11 dan SD Card* sebagai *storage* sistem dan *Wifi Dongle edimax*. Setelah merancang perangkat keras

selesai, kemudian merancang perangkat lunak yang terdiri dari sistem operasi Debian 8.6, perancangan protokol *routing* yang menggunakan protokol *routing babel*, sedangkan perancangan aplikasi *pengiriman data* menggunakan aplikasi *Client-Server pengiriman data* untuk mengirim pesan teks antar node. Dari hasil merancang perangkat keras dan perangkat lunak, menghasilkan node *manet* yang bisa berfungsi sebagai *transmitter, receiver dan router.*

1.4.2 Perancangan *node*

Seperti yang sudah disebutkan di dalam analisis kebutuhan perangkat keras, pada perancangan perangkat keras terdapat *Beaglebone Black, Micro SD SanDisk* dan *Wifi dongle edimax EW-7811Un*. Mikro *SD* berfungsi sebagai *storage Beaglebone Black* untuk menyimpan sistem operasi dan penyimpanan data. *Node Beaglebone Black* dipasang *wifi* sebagai perangkat *nirkabel* untuk membangun jaringan *Manet* yang berfungsi untuk menghubungkan dengan node lainnya.



Gambar 3.5 Beaglebone Black

1.4.3 OS *Debian 8.6 iot ARM*

Setelah pemasangan perangkat keras selesai, langkah selanjutnya adalah menginstal sistem operasi untuk *Beaglebone Black*, sistem operasi yang digunakan adalah sistem operasi *Debian 8.6*. *Debian 8.6*2016 merupakan distribusi yang mengadaptasi prinsip *keep-it-simple-stupid* dan tidak berbasis distribusi manapun. Sangat ringan dan cepat karena sistem operasi dalam keadaan kosong, dari sinilah dapat membangun dan memasang paket-paket sesuai dengan kebutuhan untuk menjalankan babel protokol.

1.4.4 Protokol *Routing Babel*

Setelah *Beaglebone Black* sudah di *installoperating systemDebian 8.6*, langkah selanjutnya adalah menginstall protokol *routing babel*. Proses *installing* protokol *routing babel* ini membutuhkan paket yang harus di *install* terlebih dahulu yang akan dijelaskan dalam bab V, Protokol routing babel merupakan protokol *routing* yang termasuk jenis *distance vector routing* protokol untuk membuat jaringan *Manet*, keunggulannya dari protokol routing ini adalah dapat di pakai pada *IPv4* maupun *IPv6 (Dualstack)*. Disamping itu, protokol *routing babel* cocok digunakan pada*Beaglebone Black* karena hanya membutuhkan persyaratan *CPU* dan memori yang kecil.

Tabel 3.1alur protokol routing *babel*

Alur babel routing protocol
- Mendeteksi semua link yang terhubung di dalam *node.* - Mencatat semua informasi link ke dalam table *routing.* - Informasi router menyimpan alamat komunikasi yang menghubungkan dengan router lain. - Router yang satu dengan yang lain saling bertukar informasi agar didapati lengkapnya isi data router. - Jika terjadi perubahan jalur jaringan, maka *router* akan segera meng*update* informasi *routing*. - Router melakukan update yangdilaksanakan bertahap tahap. - Pada saat kondisi*router*terlalu jauh dari jangkauan, waktu yang dibutuhkan menyalurkan informasi melambat. - Beritahukan Setiap link node yang terhubung di dalam *node* ke *node* tetangga

1.4.5 Aplikasi *Pengiriman data*

Konsep untuk memahami pembentukan koneksi *Client-Server* dapat diterangkan sebagai berikut:

1. *Server* membuat sebuah *socket* dengan penentuan manual alamat IP dan nomor port, yang dapat diidentifikasi dan ditemukan oleh *Client*, pada saat ini *Server* telah memasuki kondisi listening. Kondisi listening adalah keadaan dimana *Server* dalam kondisi siap untuk menerima permintaan servis dari *Client*.
2. *Client* membuat *socket*, mencari alamat *socketServer* dan kemudian menyambungkannya untuk menginisialisasi sebuah komunikasi.
3. Setelah inisialisasi dilakukan maka *Client* dan *Server* sudah bisa saling mengirimkan *data* dan menerima *data.*

Berikut ini adalah gambar mengenai hubungan antara *Server* dan beberapa *Client*.

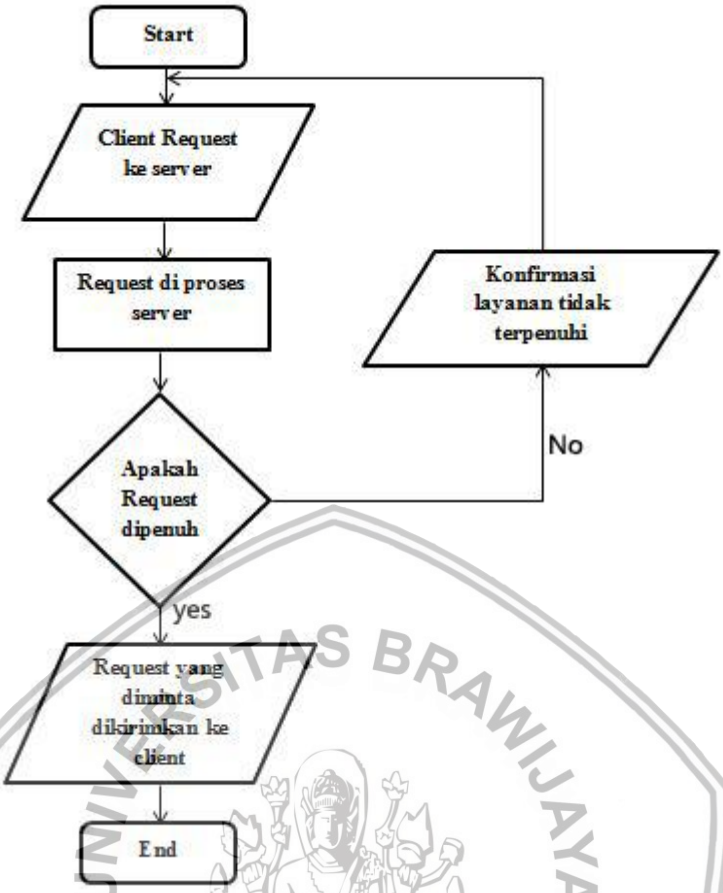
Client

server

Client

Gambar 3.6 Hubungan *Server* dengan beberapa *Client*

Diagram komunikasi *pengiriman data*mengartikan adasatu *Server* dan dua*Client*. *Client* request kemudian server menanggapi. Kemudian permintaan dari client diproses apakah persyaratan dapat dilanjutkan. Jika tidak dapat dilanjutkan maka client melakukan request ulang. Jika dapat dipenuhi permintaan server, maka server memberikan hasil permintaan server. Untuk rancangan gambar dapat dilihat dibawah ini:



Gambar 3.7 Flowchart Aplikasi pengiriman data

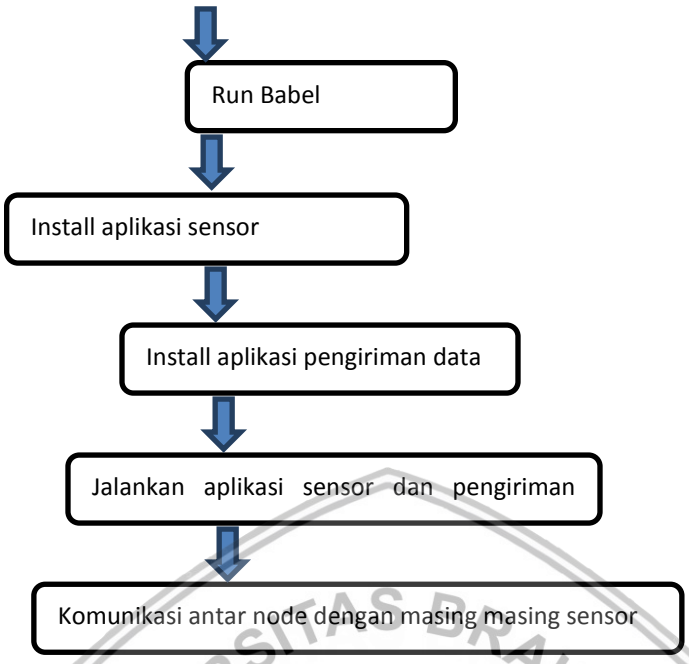
1.5 Implementasi

Implementasi merupakan tahapan penjelasan langkah langkah perancangan program sampai hasil yang akan ditampilkan pada tahapan akhir. Hal ini digambarkan pada gambar 3.8.

Perancangan Node

OS Debian 8.6

Install Babel



Gambar 3.8 Langkah implementai secara umum

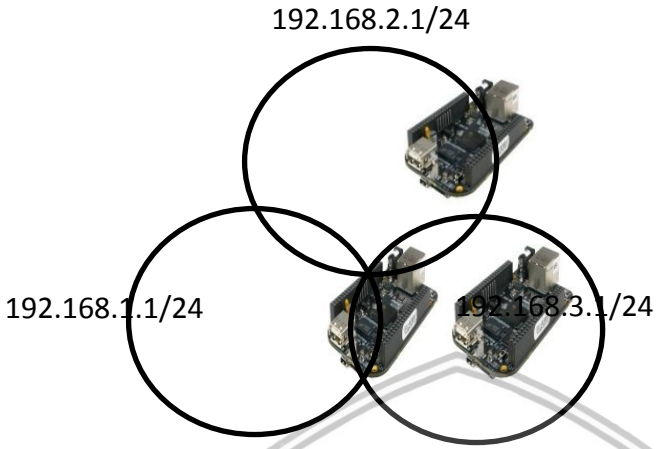
Langkah langkah implementasi secara mendetail meliputi:

1. Perancangan Node (*Beaglebone Black, Sensor LDR, DHT11, Ultrasonic, SD Card* dan *Widi Edimax).*
2. Install system operasi Debian 8.6 pada SD Card.
3. Install protokol routing *babel* .
4. Jalankan protokol routing *babel.*
5. Install aplikasi sensor pada tiap tiap *Beaglebone.*
6. Intall aplikasi pengiriman data.
7. Jalankan aplikasi sensor dan pengiriman data.
8. Pengujian aplikasi pengiriman data antara client server dan aplikasi sensor,lalu menampilkan hasil pada tiap tiap *Beaglebone.*

1.6 Skenario penelitian

Skenario penelitian ini terdiri dari 3*nodeBeaglebone Black* yang setiap *node* telah dipasang *wifidongle edimax* dengan sistem operasi *Debian 8.6*, tiap *node* terkoneksi dengan *node* lainnya dalam jangkauan *transmisi* tertentu dengan *coverage signal* 9 meter yang merupakan jarak maksimal untuk pengiriman data dari signal wifi edimax yang digunakan. Jika komunikasi berada di luar lingkup jangkauan, maka dibutuhkan node tambahan sebagai terminal jembatan penghubung, dalam hal ini node disebut sebagai terminal. *Node* bergerak bebas dan secara otomatis terkoneksi ketika mencapai signal range *wifi* masing-masing *node*. Ketika semua *node*

sudah terkoneksi satu sama lain, lakukan 3 skenario perubahan topologi untuk mengetahui efektifitas dari *Manet*, lakukan komunikasi dengan menjalankan sensor pada tiap node*Pengiriman data* untuk saling mengirim data, dari ketiga*node* tersebut harus ada salah satu *node* yang menjadi *ServerPengiriman data* untuk melayani *Client*.



Gambar 3.9 Rancangan Arsitektur

1.7 Pengujian

Pengujian komunikasi dalam penelitian agar mengetahui system dapat berjalan sesuai dengan kebutuhan. Serta menguji kehandalan jaringan *Manet* ketika nanti akan di implementasikan menggunakan *Beaglebone Black* pada daerah Perang. Adapun pengujian yang dilakukan adalah:

1.7.1 Pengujian *Self configure*

Pengujian yang pertama adalah melakukan pengujian untuk mengetahui kemampuan dari jaringan *Manet* dalam melakukan *self configure*. *Self configure* adalah kemampuan untuk mengkonfigurasi secara mandiri dari *nodeManet* untuk bergabung dengan jaringan *Manet* yang sudah ada sebelumnya sehingga dapat meneruskan paket-paket *data* yang akan dikirimkan dapat melalui *node* yang baru bergabung tersebut. Terdapat 3 skenario letak *node* dalam pengujian ini.

1.7.2 Pengujian *self healing*

Pengujian yang kedua adalah melakukan pengujian untuk mengetahui kemampuan dari jaringan *Manet* dalam melakukan *self healing*. *Self healing* adalah kemampuan dari *nodeManet* untuk mencari jalur *routing* yang baru apabila pada jalur yang dilaluinya mengalami kerusakan atau hilangnya *node* dalam jaringan. Gangguan pada jalur ini dapat terjadi karena jalur yang dilewati bermasalah. Intinya adalah jika terjadi kegagalan dalam mengirimkan paket *data*, *nodeManet* akan mencari *rute* alternatif untuk meneruskan paket yang akan dikirimkan.

1.7.3 Pengujian Komunikasi Antar *Node*

Pengujian yang ketiga adalah melakukan pengujian pada aplikasi *Pengiriman data* untuk komunikasi pengiriman data berbasis teks antar *node*. Dari tiga *node* yang ada salah satunya

akan menjadi *Server*, sedangkan *node* yang lain akan menjadi *Client* yang saling berkomunikasi melalui *Server*.

1.8 Pengambilan Kesimpulan

Kesimpulan pada bagian ini dapat dilakukan setelah tahap perancangan, implementasi, kemudian pengujian system sudah selesai. Bertujuan untuk menjawab hal-hal yang menjadi masalah pada permasalahan sebelumnya.

BAB 4 REKAYASA KEBUTUHAN

Membahas mengenai tahapan kinerja *manet* dan pengiriman data yang diambil dari sensor melalui analisis yang dibutuhkan kemudian alur perancangan perangkat lunak/keras.

4.1. Spesifikasi Lingkungan Sistem

Implementasi *Manet* dan *pengiriman data*dibagi menjadi dua bagian yakni perangkat keras(hardware) dan perangkat lunak(Software).

4.1.1 Spesifikasi Lingkungan Perangkat Keras

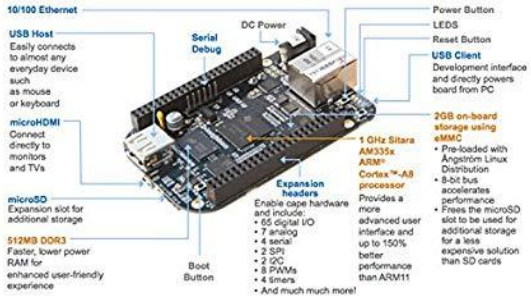
Implementasi *Manet* dan pembuatan aplikasi *pengiriman data* pada penelitian *Wireless Sensor Network(WSN)* menggunakan 3 perangkat keras yaitu *Beaglebone*, *SD Card*, dan *WifiDongle edimax*. Sensor dipasang pada tiap-tiap node yang mana saling bertukar data melalui *babel routing protocol.*

4.1.2 *Beaglebone*

Beaglebone adalah *Single Board, mini computer* yang dikembangkan oleh *Beaglebone*. Berikut adalah tabel penjelasan tentang spesifikasi hardware *Beaglebone*.

Tabel 4.1 Spesifikasi *Beaglebone*

Beaglebone	
Processor	*AM335x 1GHz ARM® Cortex-A8*
CPU	*4GB on-board flash storage*
GPU	*3D graphics*
Memory	*512MB DDR3 RAM*
Ethernet	*onboard 10/100 Ethernet RJ45 jack*
USB 2.0	*Dual USB Connector*
Video Output	*HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)*
Audio Output	*3.5mm jack, HDMI*
Onboard Storage	*SD card*
Operating System	*Debian 8.6*



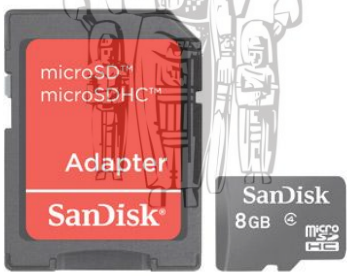
Gambar 4.1 Beaglebone

4.1.3 SD Card

SD Card merupakan Storage yang digunakan dalam penyimpanan *data* pada *Raspberry*, berikut adalah table penjelasan tentang spesifikasi hardware *SD Card*.

Tabel 4.2 Spesifikasi *SD Card*

Micro SD SanDisk 8 GB	
Kapasitas Penyimpanan	*8 GB*
Kecepatan	*Up to 30MB/second – Class 10*
Ukuran	*6.7 x 4.5 x 1.1 inci*
Type produk	*Flash memory card*



Gambar 4.2 Micro SD SanDisk 8 GB

4.1.4 *Wifi*

Wifi Edimax merupakan USB adapter berbasih nirkabel yang jangkauannya lebih baik dengan kecepatan tinggi.

Tabel 4.3 Spesifikasi *Wifi*

Wifi dongle edimax EW-7811Un

Standar	*IEEE802.11b, 802.11g, 802.11n*
Hardware Interface	*USB 2.0 Tipe A, Antena internal*
Frequensi band	*2,4000 ~ 2.4835GHz*
Data rate	- *11b: 1/2/5.5/11Mbps* - *11g: 6/9/12/24/36/48/54Mbps* - *11n (20MHz): MCS0-7 (sampai 72Mbps)* - *11n (40MHz): MCS0-7 (sampai dengan 150Mbps)*
KELEMBABAN & SUHU	*Operasi: 10 ~ 90% (Non Kondensasi)* *Penyimpanan: Max. 95% (Non Kondensasi)* *Operasi: 32 ~ 104 ° F (0 ~ 40 ° C)* *Penyimpanan: -4 ~ 140 ° F (-20 ~ 60 ° C)*
OUTPUT DAYA	*11b: 17 ± 1.5dbm* *11g: b: 15 ± 1.5dbm* *11n: 14 ± 1.5dbm*
PERSYARATAN SISTEM	*Windows XP/Vista/7* *Linux & Mac OS*

Gambar 4.3 Wifi dongle edimax EW-7811Un

Sumber :(Edimax, 2012)

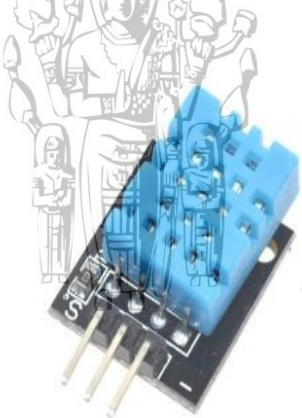
4.1.5 Sensor

1. *LDR*(Light Dependent Resistance) merupakan sensor cahaya, resistensi photoresistor menurun dengan meningkatnya intensitas cahaya. Hal ini menunjukkan sebuah fotoresistor dapat diterapkan dalam rangkaian detector cahaya.

Gambar 4.4 Sensor LDR

Sumber :(Edimax, 2012)

2. **DHT11**merupakan sensor pengukur suhu dan kelembapan dalam satu alat.Dalam sensor ini terdapat thermistor tipe NTC(Negative Temperature Coeficient) untuk mengukur suhu, sebuah sensor kelembapan tipe resisitif dan sebuah mikrokontroller 8-bit yang mengolah system dan mengirim hasill ke output.



Gambar 4.5sensor DHT11

3. Ultrasonic mengubah besaran fasis menjadi besaran listrik atau sebaliknya, berdasarkan prinsip pemantulan gelombang, lalu hasil pemantulan gelombang suara digunakan untuk menafsirkan eksistensi(jarak) pada frekuensi tertentu.

Gambar 4.6 Sensor Ultrasonic

Sumber :(Edimax, 2012)

4.2 Spesifikasi Lingkungan Perangkat Lunak

Menjelaskan Software yang dibutuhkan dalam implementasi *Manet* dan pembuatan aplikasi *chat* pada penelitian sistem komunikasi alternative bencana menggunakan teknologi *Manet* pada table 4.4.

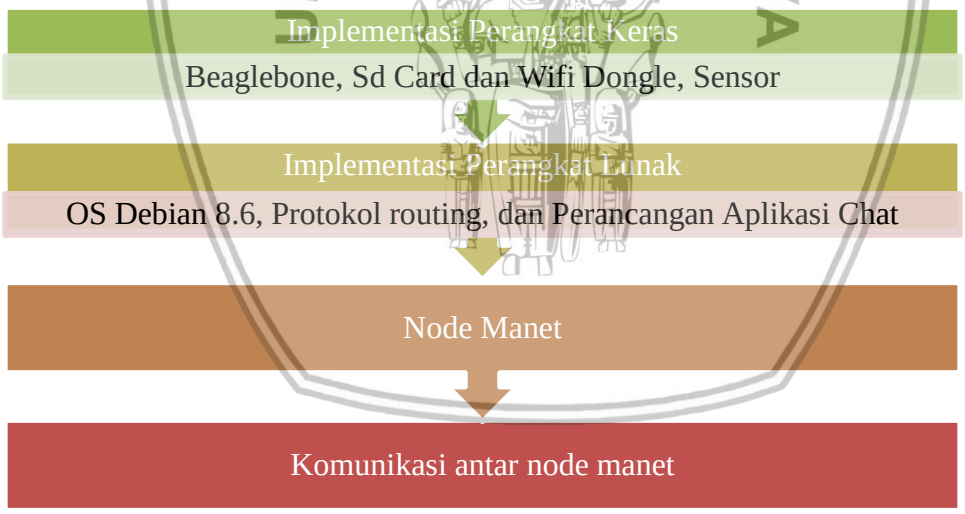
Tabel 4.4 Spesifikasi lingkungan perangkat lunak komputer

Operating system	*Debaian 13-06-2016*
Protokol *Routing*	Protokol *routingBabel*
Programming language	*Python*
Aplikasi	Sensor

BAB 5 IMPLEMENTASI

5.1 Spesifikasi Kebutuhan

Perancangan sistem secara keseluruhan, dimulai dari implementasi perangkat keras sistem, yang terdiri dari *Beaglebone, SD Card* sebagai *storage sistem* dan *Wifi Dongle edimax* sebagai perangkat *nirkabel* untuk komunikasi melalui media gelombang *radio, SD Card* dan *Wifi Dongle edimax* dipasang pada slot yang sudah disediakan pada *Beaglebone*. Setelah merancang perangkat keras selesai, kemudian implementasi perangkat lunak yang terdiri dari sistem operasi, protokol *routing*, dan aplikasi *chat*. Sistem operasi yang digunakan adalah sistem operasi *Debian 8.6*, sistem operasi ini disimpan pada *SD Card*. Kemudian, menginstall protokol *routing* pada sistem operasi *Debian 8.6*, protokol *routing* yang digunakan adalah protokol *routing babel*. Dengan adanya protokol *routing* ini menjadikan *node* bisa menggunakan *routing* secara dinamis. Setelah itu, implementasi aplikasi *chat* menggunakan aplikasi *Client-Server chat* untuk mengirim pesan teks antar *node*, pemrogaman aplikasi *chat* menggunakan bahasa pemrogaman *socketpython* yang dijalankan pada sistem operasi *Debian 8.6*. Dari hasil merancang perangkat keras dan perangkat lunak, menghasilkan *node Manet* yang bisa berfungsi sebagai *transmitter, receiver dan router.* Setelah *node Manet* terbentuk, dan membuat jaringan yang terdiri dari *node-node Manet*. Tiap *node* tersebut saling berkomunikasi menggunakan aplikasi *chat* yang tiap *node* bisa menggunakannya untuk saling mengirim pesan teks antar *node*.



Gambar 5.1 Implementasi Sistem

5.2 Instalasi *OSDebian 8.6* pada *SD Card*

1.*Download sistem operasi Beaglebone*

2. *Unzipfile*

a) Klik kanan *file* dan Pilih *"Extract file".*

b) Ikuti instruksi selanjutnya, kemudian terdapat *file.img*

3. *Downloadsoftware Win32DiskImager*.

a) *DownloadWin32DiskImager.zip*

b) *Unzip file* tersebut.

c) Akan terdapat folder baru dengan nama *Win32DiskImager-binary.*

4. Menulis *Debian 8.6ARM* pada *SD Card*

a) Pasang *SD Card* pada Labtop

b) Jalankan*fileWin32DiskImager.exe*. Terlihat seperti gambar berikut:

Gambar 5.2*Software Win32 Disk Imager*

c) Jika *SD Card* (*Device*) tidak ditemukan secara otomatis, kemudian klik drop down box dan memilihnya.

d) Pada *Imagefile* box, Pilih *fileDebian 8.6* yang telah di *Download*.

e) Klik *write*

f) Tunggu beberapa menit hingga prposes selesai, *SD Card* sudah dapat digunakan pada *Beaglebone*.

g) Tancapkan *SD Card* pada *Beaglebone,* dan nyalakan *Beaglebone* untuk booting pertama kali pada sistem operasi *Debian 8.6ARM*, masuk aplikasi putty lalu untuk *username* dan *password = root*.

5.3 Setting Wifi

Setelah instalasi sistem operasi *Debian 8.6ARM* pada *SD Card*, perlu untuk mendapatkan koneksi *internet* karena dibutuhkan paket-paket yang harus di *Download* di *internet*. Untuk melakukan proses ini harus mengatur *wifidongle edimax* terlebih dahulu. Berikut ini cara *settingwifi* untuk berkoneksi dengan *internet*:

Setelah *Booting Up, login* ke *console* menggunakan *username* dan *password = root*. Untuk mengetahui apakah *Devicewifidongle* cocok dengan *Debian 8.6*, ketikkan perintah berikut :

```
[root@beaglebone ~]# lsusb
```

Ketikkan perintah di bawah ini untuk memastikan *Wirelessinterface* telah dibuat.

```
[root@beaglebone ~]# ip link
```

Membuat *interface up* dengan *ip link set up*, untuk contohnya *sudo connmanctl*

```
root@beaglebone:~# sudo connmanctl

Error getting VPN connections: The name net.connman.vpn was not provided
by any

connmanctl> enable wifi

Error wifi: Already enabled

connmanctl> scan wifi

Scan completed for wifi

connmanctl> services
```

Setelah itu, *scan network* yang tersedia untuk koneksi internet.

Untuk pengaksesan ke koneksi internet yang tersedia kita perlu mengetahui *username* dan *passwordnya*, pada *file wpa _suplicant* kita harus memasukkan nama *wifi* dan *password* sesuai dengan konfigurasi *wifi* yang tersedia. Namun kita harus registrasi wifi jaringan yang tersedia terlebih dahulu.

```
connmanctl> agent on
Agent registered
connmanctl> connect
wifi_74da38a38ddd_63696e7461206b616d75_managed_psk
Agent RequestInput
wifi_74da38a38ddd_63696e7461206b616d75_managed_psk
  Passphrase = [ Type=psk, Requirement=mandatory ]
Passphrase? sayangkamu
Connected wifi_74da38a38ddd_63696e7461206b616d75_managed_psk
```

Start netctl untuk menghidupkan settingan *wifi* dengan menggunakan perintah

```
[root@ beaglebone~]# netctl start profile
```

Enabel koneksi ketika setiap memulai *system* dengan menggunakan perintah.

```
[root@ beaglebone ~]# netctl enable profile
```

5.3.1 Instalasi *Babel*

Untuk menginstall paket-paket *babel*, pertama kali yang dilakukan adalah *update* paket-paket *os Debian 8.6* terlebih dahulu. Ketikkan perintah :

```
[root@beaglebone ~]# sudo apt-get install babeld
```

5.3.2 Konfigurasi protokol *routingBabel*

Protokol *routingbabel* pada masing-masing *node* menggunakan program *shell script* untuk berjalan secara otomatis setiap kali *node* di jalankan. Berikut ini adalah penjelasan konfigurasi protokol *routingbabel*.

Matikan koneksi *wireless* supaya *wifi card* tidak langsung menyala saat *boot* dengan mengetikkan perintah:

```
netctl stop name.service
```

Jalankan *wifi* dongle dengan mengetikkan perintah

```
ip link set wlan0 up
```

Kemudian buat konfigurasi jaringan dengan perintah

```
iwconfig wlan0 mode ad-hoc channel 11 essid "mesh"
```

Pengalamatan IP address secara manual

```
ip addr add 192.168.x.x dev wlan0
```

Kemudian Jalankan *babel* dengan perintah

```
Sudo babeld -D wlan0
```

5.3.3 Pengalamatan *Node*

Pada pengujian ini terdapat 4 *node* yang yang ditempatkan pada posisi tertentu agar sesuai dengan topologi jaringan yang dikehendaki, Setiap *node* memiliki alamat *IP Address* yang berbeda sebagai berikut:

a. *Node*dengan alamat *IP* 192.168.1.1*Subnet Mask* 255.255.255.0
b. *Node*dengan alamat *IP*192.168.2.1*Subnet Mask* 255.255.255.0
c. *Node*dengan alamat *IP* 192.168.3.1*Subnet Mask* 255.255.255.0

5.4 SKENARIO IMPLEMENTASI

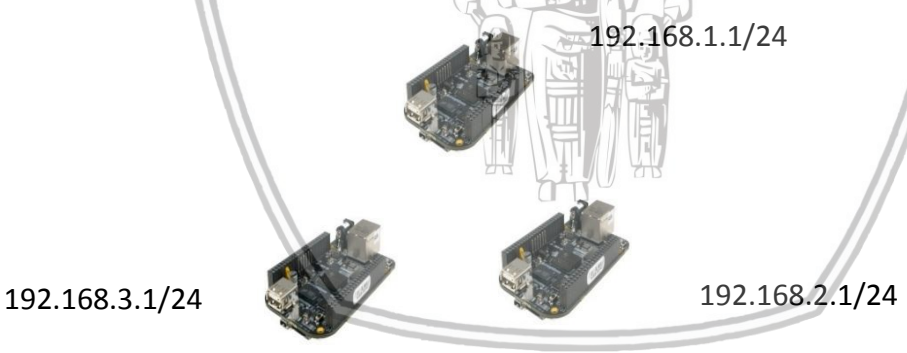
Dalam rangka membangun implementasi fisik merupakan hal yang cukup penting untuk membentuk *Manet*, karena penentuan letak tiap *node* diletakkan di beberapa posisi yang berbeda dalam jarak tertentu, pemilihan lokasi juga memperhatikan banyak faktor antara lain : tempat untuk menempatkan *node* harus memiliki sumber listrik, selain itu penempatan *node* harus terlindung dari pengaruh cuaca seperti hujan karena dapat menyebabkan kerusakan pada *node*. Terdapat 3 skenario dalam impelementasi ini.

Skenario 1:

Gambar 5.3letak *node-node* pada skenario 1

Terdapat 4 *node* dalam skenario ini, *Node* A, B, C. Semua *node* terhubung dengan signal *wifinode* lainnya. Jarak antar *node* dengan *node* lainnya 1 meter.

Skenario 2:



Gambar 5.4 Letak *node-node* pada skenario 2

Terdapat 3 *node* dalam skenario impementasi ini, *Node*C, A, dan B. Signal *node*C terhubung dengan *node*A, *node*A terhubung dengan *node*B. Ketika *node*C berkomunikas dengan *node*C harus melewati *node*A terlebih dahulu. Jarak antar *node* dengan *node* lainnya adalah 9 meter.

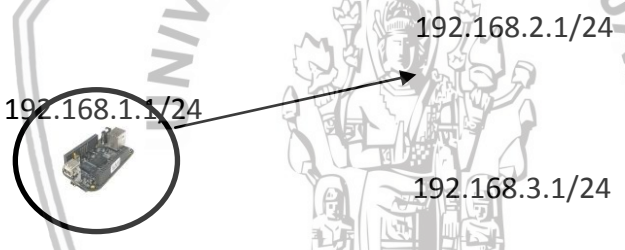
Skenario 3:

192.168.2.1/24   192.168.1.1/24

Gambar 5.5 Letak *node-node* pada skenario 3

Terdapat 3 *node* dalam skenario impementasi ini, *Node*B, C, dan A. Signal *node*B terhubung langsung dengan *node* C, *node* C terhubung dengan *node*A. Ketika *node*B berkomunikas dengan *node*A harus melewati *node*C sebagai gateway terlebih dahulu. Jarak antar *node* dengan *node* lainnya adalah 9 meter.

5.5 Skenario implementasi aplikasi *data*.



Gambar 5.6 Letak node pada implementasi *WSN*

Komunikasi yang terbentuk dari aplikasi *chat*yaknihubungan antara socket pengirim dengan penerima.telah di tentukan untuk mengidentifikasikan setiap *node* sebuah alamat IP dan nomor *port* . Hubungan koneksi antar *node* dalam implementasi ini ada dua macam, yaitu broadcast *message* dan *Privatemessage*. Pada skenario implementasi aplikasi *chatClientServer*, terdapat 3*node* yaitu *node* A, B, C dan D. *Node* A bertindak sebagai *Server*, sedangkan *node* B, C dan *node* D sebagai *Client*.

5.6 Kendala dan Solusi

Dalam penelitian ini ada beberapa kendala yang dihadapi, yaitu masalah konektivitas secara otomatis pada *routing* protokol *babel*, protokol *routingbabel* harus menggunakan *IP* dinamis supaya bisa terkoneksi secara otomatis, untuk menggunakan IP dinamis dibutuhkan protokol *AHCP (Ad-Hoc Configuration Protocol)* yang berfungsi untuk memberikan *IP* secara

dinamis pada jaringan *Manet*, sedangkan aplikasi *chatClientServer* membutuhkan *IP* statik untuk berkoneksi ke *Server* untuk melakukan komunikasi antar *node* berbasis teks.

BAB 6PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai tahapan pengujian dan pembahasan tentang sistem komunikasi pada *Manet*. Proses pengujian dilakukan melalui 2 macam pengujian yaitu pengujian *Manet* dan pengujian aplikasi *chat*. Pengujian *Manet* digunakan untuk mengetahui apakah tiap *node* bisa berkoneksi secara otomatis menggunakan protokol *routingbabel*, pada pengujian *Manet* ini dibagi lagi menjadi dua pengujian yaitu *self configure* dan *self healing*. Pengujian aplikasi *chatClientServer* digunakan untuk menguji komunikasi pengiriman pesan berbasis teks antar *node,* Dari hasil pengujian tersebut, diharapkan penelitian ini bisa diterapkan di daerah bencana yang sebenarnya.

6.1 Pengujian dan analisa

Pengujian dilakukan untuk menguji jaringan *Manet* dan aplikasi *chat* untuk sistem komunikasi yang nantinya akan diterapkan pada daerah bencana ketika sistem komunikasi lain tidak bisa digunakan, dalam pengujian-pengujian yang dilakukan membuat scenario semua komunikasi tidak dapat digunakan seperti terkena dampak bencana yang menyebabkan semua infrastruktur komunikasi mengalami kerusakan. Skenario pengujian *Manet* yang dilakukan adalah untuk melihat kemampuan dari masing-masing *nodeRaspberry* dari jaringan *Manet* dalam melakukan *self configure* dan *Self healing,* sedangkan untuk menguji aplikasi *chat* adalah dengan melakukan pengiriman pesan antar *node Client* untuk berkomunikasi.

6.1.1 Pengujian *Self configure*

Pengujian dilakukan pada bagian ini untuk melihat kemampuan dari jaringan *Manet* dalam melakukan *self configure* menggunakan protokol *routing Babel* . *Self configure* adalah kemampuan *nodeManet* melakukan *routing* dengan menggunakan protokol *routing* untuk membangun koneksi dengan *node-nodeManet* lain yang sudah ada sebelumnya sehingga dapat meneruskan paket-paket *data* yang akan dikirimkan dapat melalui *router*/*node* yang baru bergabung tersebut. Pengujian *self configure* ini penting karena ketika terjadi bencana, *node Manet* bergerak tak beraturan dengan jangkauan yang berubah-ubah, sehingga bisa mencapai suatu keadaan dimana antar *node* tidak menjangkau *signal wireless* satu sama lain, sehingga dibutuhkan *node* baru sebagai jembatan penghubung komunikasinya. Skenario pengujian ini dilakukan dengan menggunakan 3 buah *node* yang diletakkan seperti gambar 5.1.1

192.168.2.1/24

192.168.1.1/24 192.168.3.1/24

Gambar 6.1 letak *node-node* pada pengujian *self configure*

Node yang akan digunakan adalah *node* A, B, dan C. Cara yang dilakukan adalah *node* A melakukan Ping terhadap *node* C karena *node* B belum diaktifkan sehingga jaringan belum terbentuk maka pada *node* A akan muncul tampilan *request time out.* Setelah *node* B diaktifkan maka dalam beberapa saat jaringan akan terbentuk dan sistem multihop dapat berjalan sehingga *node* A dapat berkomunikasi dengan *node* C melalui *node* B. dan akan muncul pada *node* A yang melakukan Ping terhadap *node* C adalah *reply from IP node C.*

Langkah uji cobanya adalah:

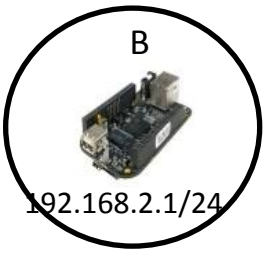
1. Letakkan *Node* A, diluar jangkauan *node* C.
2. Jalankan *babel* pada *node* A dan *node* C.
3. Kondisi awal *node* B dalam keadaan tidak aktif *(off)*
4. Kemudian aktifkan *node* B sehingga *node* A bisa terhubung dengan *node* C melalui *node* B.
5. Lakukan perintah *route* untuk mengetahui apakah semua *node* sudah terdaftar pada table *routing*.
6. Lakukan *Ping* ke semua *node* untuk mengecek apakah sudah terhubung dengan semua *node* yang terdaftar di tabel *routing*.
7. Kemampuan *node* C untuk bergabung dengan jaringan yang sudah ada inilah yang dikatakan sebagai kemampuan *self configure* dan hubungan yang terjadi menggunakan sistem *multihop.*

192.168.1.1/24 192.168.3.1/24

A **Gambar 6.2***Node* A tidak terkoneksi dengan *Node* C

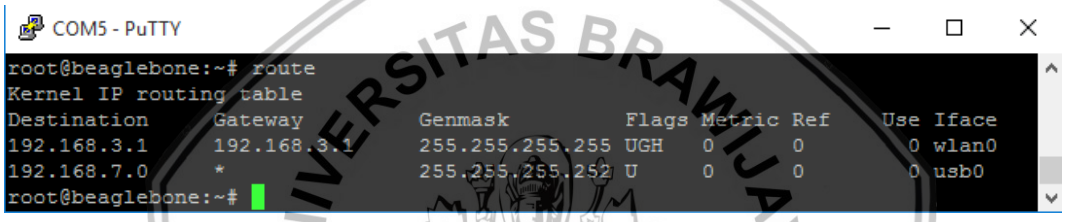
192.168.1.1/24 192.168.3.1/24

A C



Gambar 6.3*Node* A terkoneksi dengan *node* C lewat *Node* B

Pada gambar 5.2 *node* A tidak bisa melakukan komunikasi dengan *node* C, karena jalur yang dilalui belum terbentuk. Hal ini disebabkan oleh keterbatasan jarak antara *node* A dan *node* C yang berjauhan tidak mencapai signal masing-masing. Setelah *node* B ditempatkan seperti gambar 5.3, maka jalur antara *node* A dan *node* C telah terbentuk, keterbatasan jarak antara *node* A dan *node* C dapat dijembatani oleh *node* B yang ditempatkan diantara *node* A dan *node* C.



Gambar 6.4*Routing* table *node* A pengujian self configure

Perintah *route* adalah untuk melakukan *routing* pada jaringan. *Routing* adalah proses penerusan paket dari suatu *node*ke node lain agar terhubung. Informasi sumber dan tujuan disimpan pada tabel *routing*. Hasil output *route* menunjukkan bahwa *node* A terhubung dengan *node* B dan *node* C, tetapi untuk menuju *node* C harus melewati *node* B sebagai gateway.

Ping adalah perintah yang biasa digunakan untuk menguji koneksi jaringan dengan cara mengirimkan paket *data* ke*host*, lalu menampilkan waktu yang ditempuh selama proses pengiriman data tersebut. Ping adalah *Packet Internet or Inter-Network Group*. Perintah Ping menggunakan pengiriman dengan *packet Internet Protocol (IP)*, yang biasa dikenal dengan protokol *ICMP (Internet Control Message Protocol*) dengan mengirimkan *packet echo request datagram.* Setiap paket yang dikirimkan menunggu jawaban dari alamat tujuan *(destination)*. Hasil *output Ping* berisikan lamanya waktu jawaban dari alamat tujuan.

6.1.2 Pengujian *Self healing*

Skenario kedua yang dilakukan pada penelitian ini adalah untuk melihat kemampuan dari jaringan *Manet* dalam melakukan *self healing* dengan menggunakan protokol *routing Babel*. *Self healing* adalah kemampuan dari *Manet* mencari jalur *routing* yang baru apabila pada jalur yang dilaluinya terjadi kerusakan. Gangguan pada jalur ini dapat terjadi karena jalur yang dilewati bermasalah. Intinya adalah jika terjadi kegagalan dalam mengirimkan paket *data*, *node* dapat mencari jalur alternatif untuk meneruskan paket yang akan dikirimkan. Pengujian ini penting karena, ketika terjadi bencana *node* bisa mengalami kerusakan sewaktu-waktu,

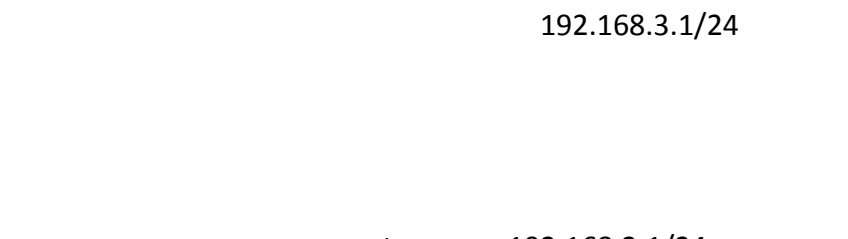
sehingga *node* harus mempunyai kemampuan untuk mencari jalur alternatif sebagai pengganti jalur yang rusak untuk meneruskan paket *data* yang dikirimkan.

Skenario yang dilakukan untuk pengujian *self healing* dalam jaringan *Manet* adalah dengan menggunakan 3 buah *nodeManet* yang telah diposisikan seperti pada gambar 6.5 dimana semua *node* berada dalam posisi aktif sehingga setiap *node* dapat melakukan Ping terhadap *node* lainnya.

Langkah-langkah yang dilakukan pada pengujian self-healing adalah:

1. Lakukan *Pingnode* A terhadap *node* C hingga mendapatkan balasan dari *node* C berupa reply from *node* C.
2. Lihat jalur yang digunakan oleh *node* A untuk berkomunikasi dengan *node* C, misal jalur yang digunakan adalah melalui:
   Node A – *node* B – *node* C seperti mematikan power pada *router* B dan perintah Ping masih terus dilakukan hingga terbentuk jalur baru seperti pada gambar 6.5*node* A-*node* D-*node* C.

Gambar 6.5 pengujian *self healing*

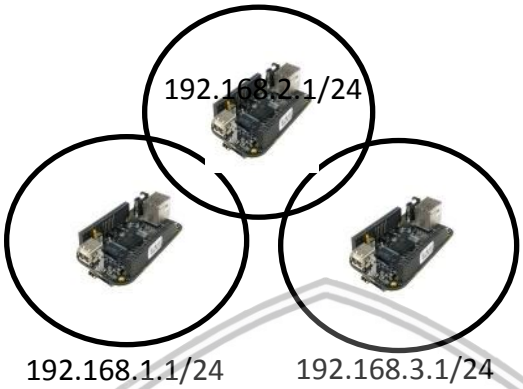


Gambar 6.6 Terjadi kerusakan pada *node* B

Pada gambar 6.5 Jalur awal yang dibentuk untuk komunikasi antara *node* A dan *node* C adalah melalui *router* B. Untuk menguji kemampuan *self healing* maka *node* B di non-aktifkan, sehingga *rute* baru terbentuk untuk komunikasi antara *node* A dan *node* C yaitu melalui jalur *node* D.

6.1.3 Pengujian Letak *NodeManet*

Skenario 1:



Gambar 6.7 letak *node-node* pada skenario 1

Pengujian *Manet* dilakukan pada penelitian ini dengan menempatkan 3 *nodeManet* dalam skenario ini, yaitu *node* A, B, dan C. Semua *node* terhubung dengan signal *wifinode* lainnya. Jarak antar *node* dengan *node* lainnya adalah 1 meter sehingga semua signal *node* saling mengenai.

```
root@beaglebone:~# iwconfig wlan0 mode ad-hoc channel 11 essid 'mesh'
root@beaglebone:~# ip addr add 192.168.3.1 dev wlan0
root@beaglebone:~# sudo babled -D wlan0
sudo: babled: command not found
root@beaglebone:~# sudo babeld -D wlan0
root@beaglebone:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
169.254.125.15  192.168.1.1     255.255.255.255 UGH   0      0        0 wlan0
192.168.1.1     192.168.1.1     255.255.255.255 UGH   0      0        0 wlan0
192.168.2.1     192.168.1.1     255.255.255.255 UGH   0      0        0 wlan0
192.168.7.0     *               255.255.255.252 U     0      0        0 usb0
root@beaglebone:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
169.254.125.15  192.168.2.1     255.255.255.255 UGH   0      0        0 wlan0
192.168.1.1     192.168.1.1     255.255.255.255 UGH   0      0        0 wlan0
192.168.2.1     192.168.2.1     255.255.255.255 UGH   0      0        0 wlan0
192.168.7.0     *               255.255.255.252 U     0      0        0 usb0
root@beaglebone:~#
```

Gambar 6.8*Routing* table *node* A pada skenario 1

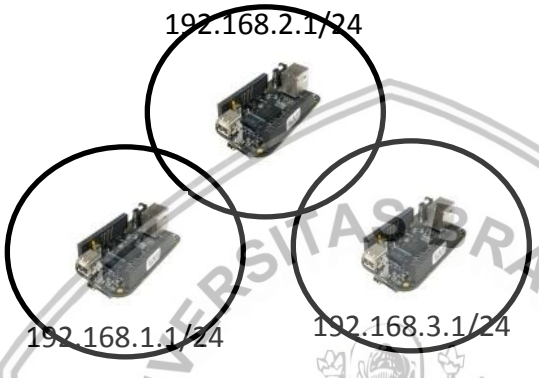
```
root@beaglebone:~# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=3.71 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=2.46 ms

--- 192.168.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 2.468/3.093/3.719/0.627 ms
^Croot@beaglebone:~# ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=9.95 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=2.23 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=3.11 ms
^C
--- 192.168.2.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 2.233/5.101/9.958/3.453 ms
root@beaglebone:~#
```

Gambar 6.9*Routing* Ping *node* A pada skenario 1

Pada gambar 6.9 Menjelaskan hasil *route* dari *node* A yang menunjukkan bahwa telah terkoneksi dengan *node* B, dan C secara langsung karena informasi gateway adalah miliknya sendiri, informasi *route* tersebut disimpan di tabel *routing*. Kemudian dilakukan Ping terhadap masing-masing *node* untuk menguji koneksi jaringan, dari hasil tersebut diketahui bahwa ketika dilakukan Ping semua terkoneksi dengan baik yang ditunjukkan dari gambar 6.10

Skenario 2:



Gambar 6.10 Letak *node-node* pada skenario 2

Pengujian *Manet* dilakukan pada penelitian ini dengan menempatkan 3 *nodeManet* dalam skenario ini, yaitu *node* A, B, dan C. *Node* A terhubung dengan signal *wifinode* B tapi tidak terhubung langsung dengan *node*C, *node* B terhubung dengan *node* A dan *node*C sehingga ketika *node* A berkomunikasi dengan *node*C harus melewati *node*B.

```
root@beaglebone:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
169.254.71.247  192.168.3.1     255.255.255.255 UGH   0      0        0 wlan0
192.168.2.1     192.168.2.1     255.255.255.255 UGH   0      0        0 wlan0
192.168.3.1     192.168.3.1     255.255.255.255 UGH   0      0        0 wlan0
192.168.7.0     *               255.255.255.252 U     0      0        0 usb0
root@beaglebone:~#
```

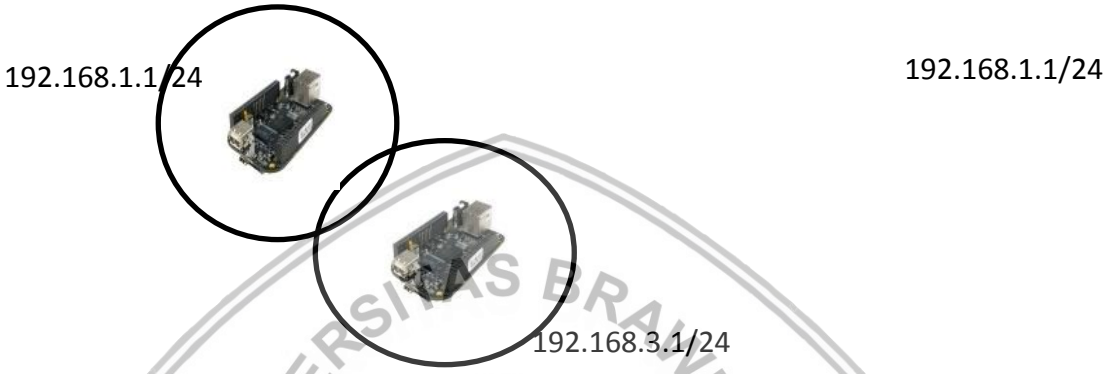
Gambar 6.11*Routing* table *node* A pada skenario 2

```
root@beaglebone:~# ping 192.168.3.1
PING 192.168.3.1 (192.168.3.1) 56(84) bytes of data.
64 bytes from 192.168.3.1: icmp_seq=1 ttl=64 time=9.99 ms
64 bytes from 192.168.3.1: icmp_seq=2 ttl=64 time=14.4 ms
64 bytes from 192.168.3.1: icmp_seq=3 ttl=64 time=2.23 ms
64 bytes from 192.168.3.1: icmp_seq=4 ttl=64 time=4.88 ms
64 bytes from 192.168.3.1: icmp_seq=5 ttl=64 time=4.83 ms
```

Gambar 6.12 Ping *node* A pada skenario 2

Pada gambar 6.11 Menjelaskan hasil *route* dari *node* A yang menunjukkan bahwa telah terkoneksi dengan *node* B, dan C. Signal *node* A tidak terhubung langsung dengan signal *node*C sehingga informasi gateway dari *node*C adalah IP dari *node* B. Informasi *route* tersebut disimpan di tabel *routing*. Kemudian dilakukan Ping terhadap masing-masing *node* untuk menguji koneksi jaringan, dari hasil tersebut diketahui bahwa ketika dilakukan Ping semua terkoneksi dengan baik yang ditunjukkan dari gambar 6.12

Skenario 3:



Gambar 6.13 Letak *node-node* pada skenario 3

Pengujian *Manet* dilakukan pada penelitian ini dengan menempatkan 3 *node*Manet dalam skenario ini, yaitu *node* A, C, dan B. *Node* A terhubung dengan signal *wifinode* C taPi tidak terhubung langsung dengan *node*B, *node* C terhubung dengan *node* A dan *node*B sehingga ketika *node* A berkomunikasi dengan *node*B harus melewati *node* C.

```
root@beaglebone:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         *               0.0.0.0         U     0      0        0 eth0
link-local      *               255.255.0.0     U     0      0        0 eth0
192.168.1.1     192.168.1.1     255.255.255.255 UGH   0      0        0 wlan0
192.168.2.1     192.168.2.1     255.255.255.255 UGH   0      0        0 wlan0
192.168.7.0     *               255.255.255.252 U     0      0        0 usb0
root@beaglebone:~#
```

Gambar 6.14*Routing* table *node* A pada skenario 3

```
root@beaglebone:~# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=11.8 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=2.20 ms
^C
--- 192.168.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 2.204/7.012/11.820/4.808 ms
root@beaglebone:~# ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=7.18 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=2.31 ms
^C
--- 192.168.2.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 2.312/4.749/7.187/2.438 ms
```

Gambar 6.15 Ping *node* A pada skenario 3

Pada gambar 6.14 Menjelaskan hasil *route* dari *node* A yang menunjukkan bahwa telah terkoneksi dengan *node* C, dan D. Signal *node* A tidak terhubung langsung dengan signal *node* D sehingga informasi gateway dari *node* D adalah IP dari *node* C. Informasi *route* tersebut disimpan di tabel *routing*. Kemudian dilakukan Ping terhadap masing-masing *node* untuk menguji koneksi jaringan, dari hasil tersebut diketahui bahwa ketika dilakukan Ping semua terkoneksi dengan baik yang ditunjukkan dari gambar 6.15

6.1.4 Pengujian Sensor

Pengujian sensor adalah menguji kestabilan sensorpada tiap*node Manet* dengan menjalankan program, hal ini dibutuhkan ketika medan wilayah tidak memadai, dan melakukan pemindaian cuaca, suhu dan kelembapan. Pengujian dilakukan pada *tiap tiap beaglebone*. Berikut adalah pengujian sensor DHT11 atau sensor suhu dan kelembapan:

```
root@beaglebone:~/temperature/Adafruit_Python_DHT/examples#
root@beaglebone:~/temperature/Adafruit_Python_DHT/examples# ./AdafruitDHT.py 11
P8_11
Temp=32.0*  Humidity=67.0%
root@beaglebone:~/temperature/Adafruit_Python_DHT/examples#
```

Gambar 6.16 Pengujian *Sensor DHT11*

Pengujian sensor LDR(Light Dependert Resistor):

```
root@beaglebone:~# python light.py
Reading         Volts
0.861538        1550.769246
0.863492        1554.285729
0.864225        1555.604410
0.541392        974.505544
0.442491        796.483523
0.384860        692.747265
0.398535        717.362648
0.427839        770.109898
0.432234        778.021985
0.433944        781.098908
0.427350        769.230777
```

Gambar 6.17 Pengujian *Senor LDR*

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil perancangan, Implementasi dan pengujian yang dilakukan, maka peneliti dapat menambil kesimpulan berikut ini:

1. Berdasarkan hasil pengujian *self configure* menggunakan protokol *routing Babel*, node dapat mengkonfigurasi otomatissehingga langsung terkoneksi dengan *node* lainnya.
2. Berdasarkan hasil pengujian *self healing*menggunakan protokol routing Babel, node harus melakukan reboot terlebih dahulu supaya bisa meneruskan paket dengan jalur routing yang baru.
3. Berdasarkan hasil pengujian letak *nodeManet*menggunakan protokol *routing Babel* menunjukkan bahwa *node* dapat berkoneksi secara mandiri sesaat setelah *startup*sistem operasi.
4. Pengujian aplikasi *pengiriman data* menggunakan metode *peer to peer* untuk melakukan komunikasi antar *node* dengan cara *broadcast* pesan ke semua *Client* kecuali *Client* pengirim.
5. Berdasarkan hasil pengujian sensor, setiap sensor mampu menunjukkan kestabilan pada tiap tiap node.

7.2 Saran

Saran dari penulis untuk perkembanganpenelitianlebih lanjut :

1. Untuk pengembangan selanjutnya, diharapkan terdapat *LCD* dan *keyboard mini* yang terpasang pada *Beaglebone* sehingga untuk menerapkan teknologi *Manet* menggunakan *Beaglebone* bisa semakin *mobile*.
2. Untuk pengembangan lebih lanjut, diharapkan menambahkan aspek *security* pada *Manet.*
3. Untuk pengembangan lebih lanjut, bisa menggunakan *IP* dinamis menggunakan *AHCP(Ad-Hoc Configuration Protocol)*
4. Untuk pengembangan lebih lanjut dapat menyediakan aplikasi client server yang mana sensor dapat diakses melalui node/client yang lain.

DAFTAR PUSTAKA

Julius Chroboczek. (2011), The Babel Routing Protocol. [online] Tersedia di: <https://tools.ietf.org/html/rfc6126> [Diakses 7 Mei 2018]

Mauliza Yefrianti. (2017). Aplikasi Group Chatting Menggunakan Manet(Mobile Ad-hoc Network). [online] Tersedia di: <https://jurnal.pcr.ac.id/index.php/jae/article/view/938> [Diakses 13 Mei 2018]

I Nyoman Budha Hartawan., Waskitho Wibisono. (2013). Mekanisme Pemeliharaan MPR Dengan Congestion Detection Dalam OLSR Pada Manet. [online] Tersedia di: <https://ojs.unud.ac.id/index.php/jik/article/view/8409> [Diakses 8 Juni 2018]

Nguyen, cole, herberg. (2013). Network management of mobile ad hoc network (*MANET*). [online] Tersedia di: <https://tools.ietf.org/html/draft-nguyen-manet-ecds-mib-02> [diakses 12 Juni 2018]

Staub. (2012). Comparative Analysis of Reactive, Proactive and Hybrid Routing Protocol in *Manet*.[online] Tersedia di: <https://www.researchgate.net/publication/267941228_Comparative_Analysis_of_Reactive_Proactive_and_Hybrid_Routing_Protocols_in_MANET> [Diakses 10 Juni 2018]

Binus. (2012). Introduction of Mobile Ad-hoc Network(*Manet)*.[online] Tersedia di:<http://binus.ac.id/malang/2017/09/introduction-of-mobile-ad-hoc-network-manet/> [Diakses 14 Juni 2018]